Coding for Combined Block–Symbol Error Correction

Ron M. Roth, *Fellow, IEEE,* and Pascal O. Vontobel, *Senior Member, IEEE*

Abstract—We design low-complexity error correction coding schemes for channels that introduce different types of errors and erasures: on the one hand, the proposed schemes can successfully deal with symbol errors and erasures, and, on the other hand, they can also successfully handle phased burst errors and erasures.

Index Terms—Decoding, generalized Reed–Solomon code, Feng–Tzeng algorithm, phased burst erasure, phased burst error, symbol erasure, symbol error.

I. INTRODUCTION

Many data transmission and storage systems suffer from different types of errors at the same time. For example, in some data storage systems the state of a memory cell might be altered by an alpha particle that hits this memory cell. On the other hand, an entire block of memory cells might become unreliable because of hardware wear-out. Such data transmission and storage systems can be modeled by channels that introduce symbol errors and block (*i.e.*, phased burst) errors, where block errors encompass several contiguous symbols. Moreover, if some side information is available, say based on previously observed erroneous behavior of a single or of multiple memory cells, this can be modeled as symbol erasures and block erasures.

In this paper, we design novel error correction coding schemes that can deal with both symbol and block errors and both symbol and block erasures for a setup as in Figure 1.

- Every small square corresponds to a symbol in $F = \mathrm{GF}(q)$, where q is an arbitrary prime power. (In applications, q is typically a small power of 2.)
- All small squares are arranged in the shape of an $m \times n$ rectangular array.
- We say that a *symbol error* happens if the content of a small square is altered. We say that a *block error* happens if one or several small squares in a column of the array are altered.
- Similarly, we say that a *symbol erasure* happens if the content of a small square is erased and we say that a *block erasure* happens if all small squares in a column of the array are erased.

We can correct such errors and erasures by imposing that the symbols in such an array constitute a codeword in some suitably chosen code $\mathbb{C}$ of length mn over F. The two main
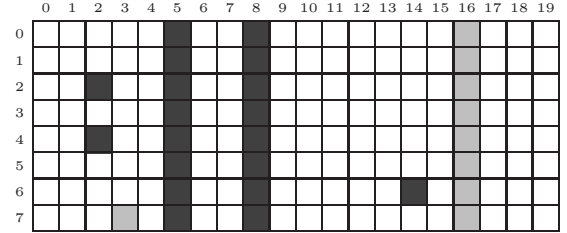
Fig. 1. Array of size $m \times n$ with symbol block errors and erasures. Here, $m = 8$, $n = 20$, and there are symbol errors at positions $(2, 2)$, $(4, 2)$, and $(6, 14)$, a symbol erasure at position $(7, 3)$, block errors in columns 5 and 8, and a block erasure in column 16.

ingredients of the code $\mathbb{C}$ that is proposed in this paper are, on the one hand, a matrix H_{in} of size $m \times (mn)$ over F, and, on the other hand, a code $\mathcal{C}$ of length n over F. Namely, an array forms a codeword in $\mathbb{C}$ if and only if every row of the array is a codeword in $\mathcal{C}$ once the n columns have been transformed by n different bijective mappings $F^m \to F^m$ derived from the matrix H_{in}. The resulting error-correcting coding scheme has the following salient features:

- It can be seen as a concatenated coding scheme, however with two somewhat non-standard features. First, multiple inner codes are used (one for every column encoding), and, second, all these inner codes have rate one.
- One can identify a range of code parameters for $\mathbb{C}$ for which (to the best of our knowledge) the resulting redundancy improves upon the best known.
- One can devise efficient decoders for combinations of symbol and block errors and erasures most relevant in practical applications. In particular, these decoders are more efficient than a corresponding decoder for a suitably chosen generalized Reed–Solomon (GRS) code of length mn over F, assuming such a GRS code exists in the first place. (Finding efficient decoders for the general case is still an open problem.)

A. Paper Overview

The paper starts in Section II by considering a simplified version of the above error and erasure scenario and of the above-mentioned code construction. Namely, in this section we consider only block errors and erasures, *i.e.*, no symbol errors or erasures. Moreover, an $m \times n$ array forms a codeword if and only if every row is a codeword in some code $\mathcal{C}$ of length n (*i.e.*, there are no bijective mappings applied to the columns); in other words, the array code considered is simply an m-level interleaving of $\mathcal{C}$. Our main purpose of Section II is laying out some of the ideas and tools that will be used in

subsequent sections; in particular, it is shown how one can take advantage of the *rank* of the error array in order to increase the correction capability of the array code. Nevertheless, the discussion in Section II-C may be of independent interest in that it provides a simplified analysis of the decoding error probability of interleaved GRS codes when used in certain (probabilistic) channel models.

We then move on to Section III, which is the heart of the paper and which gives all the details of the above-mentioned code construction and compares it with other code constructions. Finally, Section IV discusses a variety of decoders for the proposed codes.

B. Related Work

The idea of exploiting the rank of the error array when decoding interleaved codes was presented by Metzner and Kapturowski in [22] and by Haslach and Vinck in [12], [13]. Therein, the code $\mathcal{C}$ is chosen to be a linear $[n, k, d]$ code over F, and, clearly, any combination of block errors can be corrected as long as their number does not exceed $(d-1)/2$. In [22] and [12], it was further assumed that the set of nonzero columns in the (additive) $m \times n$ error array E over F is linearly independent over F; namely, the rank of E (as a matrix over F) equals the number of block errors. It was then shown that under this additional assumption, it is possible to correct (efficiently) any pattern of up to $d - 2$ block errors. Essentially, the linear independence allows to easily locate the nonzero columns in E, and from that point onward, the problem reduces to that of erasure decoding. A generalization to the case where the nonzero columns in E are not necessarily full-rank was discussed in [13]; we will recall the latter result in mode detail in Section II-A.

The case where the constituent code $\mathcal{C}$ is a GRS code has been studied in quite a few papers, primarily in the context where the contents of each block error is assumed to be uniformly drawn from F^m. In [3], Bleichenbacher *et al.* identified a threshold, $(m/(m+1))(d-1)$, on the number of block errors, below which the decoding failure probability approaches 0 as d goes to infinity and n/q goes to 0. A better bound on the decoding error probability was obtained by Kurzweil *et al.* [18] and Schmidt *et al.* [27]. See also Brown *et al.* [5], Coppersmith and Sudan [6], Justesen *et al.* [14], Krachkovsky and Lee [17], and Wachter–Zeh *et al.* [29].

Turning to the main coding problem studied in this paper—namely, handling combinations of symbol errors and block errors—a general solution was given by Zinov'ev [32] and Zinov'ev and Zyablov [33], using concatenated codes and their generalizations. Specifically, when using an (ordinary) concatenated code, the columns of the $m \times n$ array are set to be codewords of a linear $[m, \mathsf{k}, \mathsf{d}]$ inner code over F, and each of these codewords is the result of an encoding of a coordinate of an outer codeword of a second linear $[n, \mathsf{K}, \mathsf{D}]$ code over $\mathrm{GF}(q^{\mathsf{k}})$. It follows from the analysis in [32] and [33] that any error pattern of up to ϑ symbol errors and τ block errors can be correctly decoded, whenever

$$2\vartheta + 1 \leq \mathsf{d}(\mathsf{D} - 2\tau) .$$

Furthermore, such error patterns can be efficiently decoded, provided that the inner and outer codes have efficient bounded-distance error–erasure decoders.

Note that (in the nontrivial case) when $\vartheta > 0$, the rate of the inner code must be (strictly) smaller than 1. This, in turn, implies that the overall redundancy of the concatenated code has to grow (at least) linearly with n. A generalized concatenated (GC) code allows to circumvent this impediment. We briefly describe the approach, roughly following the formulation of Blokh and Zyablov [4]. Given a number τ of block errors and a number ϑ of symbol errors that need to be corrected, let the integer sequences

$$1 = \mathsf{d}_0 < \mathsf{d}_1 < \cdots < \mathsf{d}_v ,$$
$$\mathsf{D}_0 \geq \mathsf{D}_1 \geq \cdots \geq \mathsf{D}_v$$

satisfy, for every $i = 0, 1, \ldots, v$,

$$2\vartheta + 1 \leq \mathsf{d}_i(\mathsf{D}_i - 2\tau) \tag{1}$$

(thus, $\mathsf{D}_0 \geq 2(\tau + \vartheta) + 1$ and $\mathsf{D}_v \geq 2\tau + 1$). For $i = 1, 2, \ldots, v$, let H_i be an $\mathsf{r}_i \times m$ parity-check matrix of a linear $[m, \mathsf{k}_i = m - \mathsf{r}_i, \mathsf{d}_i]$ over F. We further assume that these codes are (strictly) nested, so that H_{i-1} forms a proper $\mathsf{r}_{i-1} \times m$ sub-matrix of H_i; the matrix formed by the remaining $\mathsf{r}_i - \mathsf{r}_{i-1}$ rows of H_i will be denoted by $\partial \mathsf{H}_i$ (we formally define $\mathsf{r}_0 = 0$, along with setting H_0 to be an "empty" $\mathsf{r}_0 \times n$ matrix). Then an $m \times n$ array over F is a codeword of the generalized concatenated code, if and only if the following two conditions hold:

(G1) For $i = 1, 2, \ldots, v$, the (partial) syndromes of the columns with respect to the partial parity-check matrix $\partial \mathsf{H}_i$ form a codeword of a code of length n over $F^{\mathsf{r}_i - \mathsf{r}_{i-1}}$ with minimum distance D_{i-1}.

(G2) The columns of the array form a codeword of a code of length n over F^m with minimum distance D_v.

(Note that condition (G2) could be incorporated into condition (G1) by extending the latter to $i = v + 1$, with H_{v+1} taken as an $m \times m$ (nonsingular) parity-check matrix of the trivial code $\{\mathbf{0}\}$. Ordinary concatenated codes correspond to the case where $v = 1$ and D_0 is the "minimum distance" ($> n$) of the trivial code.) It follows from [32] and [33] that the above array code construction has an efficient decoder that corrects any pattern of up to τ block errors and up to ϑ symbol errors. See also [1], [15], [25], [31], and the survey [7].

Recently, Blaum *et al.* [2] have proposed new erasure-correcting codes for combined block–symbol error patterns. The advantage of their scheme is having the smallest possible redundancy (equaling the largest total number of symbols that can be erased) and an efficient *erasure* decoding algorithm. However, the parameters of their constructions are rather strongly limited: first, the array size is typically much smaller than q (and, in one embodiment, must in fact be smaller than $\log_2 q$), and, secondly, verifying whether the construction actually works for given parameters becomes intractable, unless the number of block erasures or the number of symbol erasures is very small.

In Section III-C we will compare our coding scheme with the most relevant of the above-mentioned coding schemes.

C. Notation

This subsection lists the notation that will be used throughout the paper. More specialized notation will be introduced later on when needed.

For integers a and b, we denote by $\langle a, b \rangle$ the set of integers $\{a, a+1, a+2, \ldots, b-1\}$, and $\langle b \rangle$ will be used as a shorthand notation for $\langle 0, b \rangle$. Entries of vectors will be indexed starting at 0, and so will be the rows and columns of matrices. For a vector $\boldsymbol{u} \in F^n$ and a subset $W \subseteq \langle n \rangle$, we let $(\boldsymbol{u})_W$ be the sub-vector (in $F^{|W|}$) of $\boldsymbol{u}$ that is indexed by W. The support of $\boldsymbol{u}$ will be denoted by $\mathsf{supp}(\boldsymbol{u})$. We extend these definitions to any $m \times n$ matrix E over F, with $(E)_W$ denoting the $m \times |W|$ sub-matrix of E that is formed by the columns that are indexed by W. Column j of E will be denoted by E_j, and $\mathsf{supp}(E)$ will stand for the column support of E, namely, the set of indexes j for which $E_j \neq \boldsymbol{0}$. The linear subspace of F^m that is spanned by the columns of E will be denoted by $\mathsf{colspan}(E)$.

The ring of polynomials in the indeterminate x over F will be denoted by $F[x]$, and the ring of bivariate polynomials in y and x over F will be denoted by $F[y, x]$.[1] For a nonzero bivariate polynomial $\varphi(y, x) = \sum_i \varphi_i(y)x^i$ in $F[y, x]$, we will let $\deg_x \varphi(y, x)$ stand for the x-degree of $\varphi(y, x)$, namely, the largest index i for which $\varphi_i(y) \neq 0$. The y-degree is defined in a similar manner. The notation $F_{m,n}(y, x)$ will stand for the set of all bivariate polynomials $\varphi(y, x) \in F[y, x]$ with $\deg_y \varphi(y, x) < m$ and $\deg_x \varphi(y, x) < n$. For an element $\xi \in F$, we denote by $\mathrm{T}_m(y; \xi)$ the polynomial $\sum_{i \in \langle m \rangle} \xi^i y^i$.

With any $m \times n$ matrix $E = (e_{h,j})_{h \in \langle m \rangle, j \in \langle n \rangle}$ over F, we associate the bivariate polynomial

$$E(y, x) = \sum_{h \in \langle m \rangle, j \in \langle n \rangle} e_{h,j} y^h x^j$$

in $F_{m,n}(y, x)$ (namely, the powers of y index the rows and the powers of x index the columns). With each column j of E we associate the univariate polynomial $E_j(y) = \sum_{h \in \langle m \rangle} e_{h,j} y^h$; thus, $E(y, x) = \sum_{j \in \langle n \rangle} E_j(y) x^j$.

II. SIMPLIFIED CODE CONSTRUCTION

In this section we consider the simplified scenario mentioned in Section I-A. Namely, we consider only block errors and erasures, *i.e.*, no symbol errors or erasures. Moreover, an $m \times n$ array forms a codeword of length mn if and only if every row is a codeword in some prescribed code $\mathcal{C}$ with parameters $[n, k, d]$ (*i.e.*, there are no bijective mappings applied to the columns); equivalently, the array code considered is simply an m-level interleaving of $\mathcal{C}$. If $\mathcal{C}$ is specified by an $(n-k) \times n$ parity-check matrix H, then the syndrome matrix S is defined to be the $m \times (n-k)$ matrix $S = YH^\mathsf{T}$, where the $m \times n$ matrix

$$Y = \Gamma + E$$

over F represents the read out (or received) message, where the $m \times n$ matrix Γ over F represents the stored (or transmitted) codeword, and where the $m \times n$ matrix E over F represents the alterations that happen to Γ over time (or during transmission). Note that our formalism treats erasures like errors, with the side information $K \subseteq \langle n \rangle$ telling us their location.

The subsections of this section are structured as follows. In Section II-A we study the error correction capabilities of the interleaved array code, where $\mathcal{C}$ is any linear $[n, k, d]$ code over F. Then, in Section II-B, we present an efficient decoder for the special case where $\mathcal{C}$ is a GRS code. Finally, in Section II-C, we present an application of the efficient decoder of Section II-B for the probabilistic decoding of the array code under the assumption that the block errors are uniformly distributed over F^m.

A. Block Errors and Erasures with Rank Constraints

We start with Theorem 2 below that generalizes the results of [22] and [12] to the case where the set of nonzero columns of the $m \times n$ error array E are not necessarily linearly independent. Note that this theorem was already stated (without proof) in the one-page abstract [13] for the error-only case (*i.e.*, no block erasures). We include the proof of the theorem not just for the sake of completeness, but also because the proof technique will be useful in Section III we well.

Toward proving this theorem, the following lemma will be helpful.

Lemma 1 *Let $\mathcal{C}$ be a linear $[n, k, d]$ code over F and let Z be a nonzero $m \times n$ matrix over F such that each row in Z is a codeword of $\mathcal{C}$. Then*

$$|\mathsf{supp}(Z)| - \mathsf{rank}(Z) \geq d - 1 .$$

Proof: Write $J = \mathsf{supp}(Z)$, $t = |J|$, and $\mu = \mathsf{rank}(Z)$, and apply the Singleton bound to the linear $[t, \mu, \geq d]$ code over F that is spanned by the rows of $(Z)_J$. ∎

Theorem 2 *Let $\mathcal{C}$ be a linear $[n, k, d]$ code over F and let H be an $(n-k) \times n$ parity-check matrix of $\mathcal{C}$ over F. Fix K to be a subset of $\langle n \rangle$ of size r. Given any $m \times (n-k)$ (syndrome) matrix S over F, there exists at most one $m \times n$ matrix E over F that has the following properties:*

(i) $S = EH^\mathsf{T}$ *, and—*
(ii) *writing $\overline{K} = \langle n \rangle \setminus K$, the values $t = |\mathsf{supp}((E)_{\overline{K}})|$ and $\mu = \mathsf{rank}((E)_{\overline{K}})$ satisfy*

$$2t + r \leq d + \mu - 2 . \tag{2}$$

Proof: We consider first the case where K is empty. The proof is by contradiction. So, assume that E and $\hat{E}$ are two distinct $m \times n$ matrices over F that satisfy conditions (i)–(ii). Write

$$t_{\max} = \max\left\{|\mathsf{supp}(E)|, |\mathsf{supp}(\hat{E})|\right\} ,$$
$$\mu_{\max} = \max\left\{\mathsf{rank}(E), \mathsf{rank}(\hat{E})\right\} ,$$

---

[1] We prefer the ordering y, x over x, y because the powers of y and the powers of x will be associated with, respectively, the rows and columns of $m \times n$ matrices like E.

and define

$$J = \mathsf{supp}(E) \ ,$$
$$\hat{J} = \mathsf{supp}(\hat{E}) \ ,$$
$$Q = \mathsf{supp}(E) \cap \mathsf{supp}(\hat{E}) \ .$$

Consider the array $Z = E - \hat{E}$. By condition (i) we get that every row in Z is a codeword of $\mathcal{C}$. Now,

$$|\mathsf{supp}(Z)| \le |J| + |\hat{J}| - |Q| \ ,$$
$$\mathsf{rank}(Z) \ge \mu_{\max} - |Q| \ ,$$

and, so,

$$|\mathsf{supp}(Z)| - \mathsf{rank}(Z) \le |J| + |\hat{J}| - \mu_{\max}$$
$$\le 2t_{\max} - \mu_{\max}$$
$$\le d - 2 \ ,$$

where the last inequality follows from condition (ii). Hence, by Lemma 1 we conclude that $Z = 0$, namely, $E = \hat{E}$, which is a contradiction to the initial assumption.

Next, we consider the case where $r = |K| > 0$. First, note that condition (ii) implies that $r \le d + \mu - 2 - 2t \le d - t - 2$; in particular, every subset of r columns in H is linearly independent. By applying elementary linear operations to the rows of H, we can assume without loss of generality that the first r rows of $(H)_K$ contain the identity matrix, whereas the remaining $n-k-r$ rows in $(H)_K$ are all-zero. Let $\tilde{H}$ be the $(n-k-r) \times (n-r)$ matrix which consists of the last $n-k-r$ rows of $(H)_{\overline{K}}$: the matrix $\tilde{H}$ is a parity-check matrix of the linear $[n-r, k]$ code $\tilde{\mathcal{C}}$ over F obtained by puncturing $\mathcal{C}$ on the positions that are indexed by K. Let $\tilde{S}$ be the $m \times (n-k-r)$ matrix which consists of the last $n-k-r$ columns of S. We have

$$\tilde{S} = (E)_{\overline{K}} \tilde{H}^{\mathsf{T}} \ . \tag{3}$$

Replacing (i) by (3) and $\mathcal{C}$ by $\tilde{\mathcal{C}}$, we have reduced to the case where K is empty. The result follows by recalling that the minimum distance of $\tilde{\mathcal{C}}$ is at least $d - r$. $\blacksquare$

The proof of Theorem 2 in [22] and [12], which was for the special case $r = 0$ and $\mu = t = |\mathsf{supp}(E)|$, was carried out by introducing an efficient algorithm for decoding up to $t \le d-2$ errors. In that algorithm, Gaussian elimination is performed on the columns of S, resulting in an $m \times (n-k)$ matrix SP^{T}, for some invertible $(n-k) \times (n-k)$ matrix P over F, such that the first μ columns in SP^{T} form a linearly independent set while the last $n-k-\mu$ columns in SP^{T} are all-zero. (As a matter of fact, through this Gaussian elimination, one finds the value of μ.) From condition (i) we then get that

$$SP^{\mathsf{T}} = E(PH)^{\mathsf{T}} \ . \tag{4}$$

Let H' be the $(n-k-\mu) \times n$ matrix that is formed by the last $n - k - \mu$ rows of PH. It follows from (4) that the columns of H' that are indexed by $\mathsf{supp}(E)$ must be all-zero. Furthermore, all the remaining columns in H' must be nonzero, or else we would have $\mu + 1 < d$ linearly dependent columns in H. It follows that $\mathsf{supp}(E)$ is the unique subset $U \subseteq \langle n \rangle$ of size μ such that $(H')_U$ is all-zero. Once the decoder identifies $\mathsf{supp}(E)$, the entries of E can be found

by solving linear equations. This decoding algorithm can be generalized to handle the case where $r = |K| > 0$, in the spirit of the last part of the proof of Theorem 2: replace E, H, and S by $(E)_{\overline{K}}$, $\tilde{H}$, and $\tilde{S}$, respectively.

As pointed out in [13], when K is empty and the difference $t - \mu = |\mathsf{supp}(E)| - \mathsf{rank}(E)$ is assumed to be equal to some nonnegative integer b, then the decoding algorithm of [22] and [12] can be generalized into finding a subset $U \subseteq \langle n \rangle$ of size $t \le (d+\mu)/2 - 1$ such that $\mathsf{colspan}\big((H')_U\big) \ (\subseteq F^{n-k-\mu})$ has dimension b. Letting V be a subset of U of size b such that $\mathsf{rank}\big((H')_V\big) = b$, the subset $W = U \setminus V$ will then point at $t - b = \mu$ columns of E that form a basis of $\mathsf{colspan}(E)$; these μ columns, in turn, are flagged as μ erasures. We will then be left with $|\mathsf{supp}(E) \setminus W| = t - \mu = b$ nonzero columns in E which are yet to be located, but these can be found by applying to the received array, row by row, any bounded-distance error–erasure decoder $\mathcal{D}$ for $\mathcal{C}$. Indeed, since

$$2b + \mu = 2t - \mu \le d - 2 \ ,$$

such a decoder can uniquely recover E given the set W of erasure locations. The extension to $r = |K| > 0$ is straightforward.

Note, however, that even when we are allowed to apply the decoder $\mathcal{D}$ at no computational cost, we do not know how to find the subset U efficiently as b becomes large. In fact, if b is large and μ is small, we may instead enumerate over the set W which indexes a basis of $\mathsf{colspan}(E)$, then use $\mathcal{D}$ to reconstruct a candidate for E, and finally verify that we indeed have $\mathsf{rank}(E) = \mathsf{rank}\big((E)_W\big) = \mu$.

B. The GRS Case

In this section, we present an efficient decoder for finding the error matrix E under the conditions of Theorem 2, for the special case where $\mathcal{C}$ is a generalized Reed–Solomon (GRS) code (this decoder will then be used as a subroutine in one of the decoders to be presented in Section IV). Specifically, hereafter in this section, we fix $\mathcal{C}$ to be an $[n, k, d=n-k+1]$ GRS code $\mathcal{C}_{\mathrm{GRS}}$ over F with a parity-check matrix

$$H_{\mathrm{GRS}} = \big(\alpha_j^i \big)_{i \in \langle d-1 \rangle, j \in \langle n \rangle} \ ,$$

where $\alpha_0, \alpha_1, \ldots, \alpha_{n-1}$ are distinct nonzero elements of F.

Let $E = (e_{h,j})_{h \in \langle m \rangle, j \in \langle n \rangle}$ be an $m \times n$ (error) matrix over F, and let K and J be disjoint subsets of $\langle n \rangle$ such that

$$J \subseteq \mathsf{supp}(E) \subseteq K \cup J$$

(the set J indexes the erroneous columns and K indexes the erasure locations). For the matrix E, define the syndrome array as the following $m \times (d-1)$ matrix S (equivalently, the bivariate polynomial $S(y, x) \in F_{m,d-1}(y, x)$):

$$S = EH_{\mathrm{GRS}}^{\mathsf{T}}$$

(compare with condition (i) in Theorem 2). In addition, define the error-locator polynomial $\Lambda(x)$ and the erasure-locator

polynomial $\mathrm{M}(x)$ by

$$\Lambda(x) = \prod_{j \in J}(1 - \alpha_j x) \,,$$

$$\mathrm{M}(x) = \prod_{j \in K}(1 - \alpha_j x) \,,$$

respectively. Also, let the modified syndrome array be the unique $m \times (d-1)$ array σ over F (namely, the unique bivariate polynomial $\sigma(y, x)$ in $F_{m,d-1}(y, x)$) that satisfies the polynomial congruence

$$\sigma(y, x) \equiv S(y, x)\,\mathrm{M}(x) \pmod{x^{d-1}} \,.$$

Finally, the (bivariate) error-evaluator polynomial $\Omega(y, x) = \sum_{h \in \langle m \rangle} \Omega_h(x)y^h$ is defined by

$$\Omega_h(x) = \sum_{j \in K \cup J} e_{h,j} \prod_{j' \in (K \cup J)\setminus\{j\}} (1 - \alpha_{j'}x) \,, \quad h \in \langle m \rangle \,.$$

Lemma 3 *Write $t = |J|$, $r = |K|$, and $\mu = \mathrm{rank}\big((E)_J\big)$, and suppose that*

$$2t + r \leq d + \mu - 2$$

(see (2)). Let $\lambda(x)$ be a polynomial in $F[x]$ and $\omega(y, x) = \sum_{h \in \langle m \rangle} \omega_h(x)y^h$ be a bivariate polynomial (of y-degree less than m) in $F[y, x]$ such that the following conditions are satisfied:

(P1) $\sigma(y, x)\lambda(x) \equiv \omega(y, x) \pmod{x^{d-1}}$, *and—*
(P2) $\deg \lambda(x) \leq (d+\mu-r)/2 - 1$ *and*
    $\deg_x \omega(y, x) < (d+\mu+r)/2 - 1$.

Then there is a polynomial $u(x) \in F[x]$ such that

$$\lambda(x) = \Lambda(x)u(x) \,,$$

$$\omega(y, x) = \Omega(y, x)u(x) \,.$$

Proof: First, by the key equation of GRS decoding, it is known that (P1)–(P2) are satisfied for $\lambda(x) = \Lambda(x)$ and $\omega(y, x) = \Omega(y, x)$ (see, for example, [24, Section 6.3 and pp. 207–208]).

Now, let $\lambda(x)$ and $\omega(y, x)$ satisfy (P1)–(P2), fix ℓ to be any index in J, and let U be a subset of J of size μ such that $\ell \in U$ and $\mathrm{rank}\big((E)_U\big) = \mu$ (recall that $J \subseteq \mathrm{supp}(E)$ and, so, $E_\ell \neq \mathbf{0}$). Let $\boldsymbol{a} = (a_h)_{h \in \langle m \rangle}$ be a row vector in F^m such that $(\boldsymbol{a}E)_U$ is nonzero on—and only on—position ℓ.

Let $\hat{J}$ be the smallest subset of $\langle n \rangle$ such that

$$\hat{J} \subseteq \mathrm{supp}(\boldsymbol{a}E) \subseteq K \cup \hat{J} \,;$$

note that $\ell \in \hat{J}$ and that $\hat{J} \subseteq \{\ell\} \cup (J \setminus U)$ and, so, $|\hat{J}| \leq$

$t - \mu + 1$. Define

$$\hat{\Lambda}(x) = \prod_{j \in \hat{J}}(1 - \alpha_j x) \,,$$

$$\hat{\Omega}(x) = \frac{1}{\prod_{j'' \in U \setminus \{\ell\}}(1 - \alpha_{j''}x)} \cdot \sum_{h \in \langle m \rangle} a_h \Omega_h(x)$$

$$= \sum_{j \in K \cup \hat{J}} (\boldsymbol{a}E_j) \prod_{j' \in (K \cup \hat{J})\setminus\{j\}} (1 - \alpha_{j'}x) \,,$$

$$\hat{\sigma}(x) = \sum_{h \in \langle m \rangle} a_h \sigma_h(x) \,,$$

$$\hat{\omega}(x) = \sum_{h \in \langle m \rangle} a_h \omega_h(x) \,.$$

Observing that $\hat{\sigma}(x)$ is the modified syndrome polynomial that corresponds to the row vector $\boldsymbol{a}E$, we get from the key equation of GRS decoding that

$$\hat{\Lambda}(x)\hat{\sigma}(x) \equiv \hat{\Omega}(x) \pmod{x^{d-1}} \,. \tag{5}$$

Moreover,

$$\gcd(\hat{\Lambda}(x), \hat{\Omega}(x)) = 1 \,, \tag{6}$$

$$\deg \hat{\Lambda}(x) = |\hat{J}| \qquad \leq t - \mu + 1 \qquad \leq \frac{d-\mu-r}{2} \,, \tag{7}$$

$$\deg \hat{\Omega}(x) < |K| + |\hat{J}| \leq r + t - \mu + 1 \leq \frac{d-\mu+r}{2} \,. \tag{8}$$

Multiplying both sides of (5) by $\lambda(x)$ we obtain

$$\hat{\Lambda}(x)\lambda(x)\hat{\sigma}(x) \equiv \lambda(x)\hat{\Omega}(x) \pmod{x^{d-1}} \,.$$

On the other hand, by (P1) we have

$$\lambda(x)\hat{\sigma}(x) \equiv \hat{\omega}(x) \pmod{x^{d-1}} \,,$$

and so, combining the last two congruences, we get

$$\hat{\Lambda}(x)\hat{\omega}(x) \equiv \lambda(x)\hat{\Omega}(x) \pmod{x^{d-1}} \,.$$

Now, from (P2) and (7)–(8) it follows that the degrees of the products on both sides of the last congruence are less than $d-1$. Hence, this congruence is actually a polynomial equality:

$$\hat{\Lambda}(x)\hat{\omega}(x) = \lambda(x)\hat{\Omega}(x) \,.$$

Thus, from (6) we get that $\lambda(x)$ is divisible by $\hat{\Lambda}(x)$; in particular, $\lambda(x)$ is divisible by $1 - \alpha_\ell x$. Ranging now over all ℓ in J, we conclude that $\lambda(x)$ can be written as $\Lambda(x)u(x)$ for some polynomial $u(x) \in F[x]$.

Finally, from (P1) and the key equation we obtain

$$\omega(y, x) \equiv \sigma(y, x)\lambda(x)$$
$$\equiv \sigma(y, x)\Lambda(x)u(x)$$
$$\equiv \Omega(y, x)u(x) \pmod{x^{d-1}} \,,$$

from which the equality $\omega(y, x) = u(x)\Omega(y, x)$ follows again by computing degrees: from (P2) we get

$$\deg_x \omega(y, x) < \frac{d + \mu + r}{2} - 1 \leq d - 1$$

and

$$\begin{aligned}
\deg_x &\big(\Omega(y,x)u(x)\big)\\
&= \big(\deg\lambda(x) - \deg\Lambda(x)\big) + \deg_x\Omega(y,x)\\
&< \deg\lambda(x) + r\\
&\le \frac{d+\mu+r}{2} - 1\\
&\le d-1 \,.
\end{aligned}$$

This completes the proof. ∎

Fix J and K to be *disjoint* subsets of $\langle n\rangle$, write $t = |J|$ and $r = |K|$, and let E be an $m \times n$ matrix of rank μ over F such that $J \subseteq \mathsf{supp}(E) \subseteq K \cup J$. Define

$$S^{(J)} = (E)_J\big((H_{\mathrm{GRS}})_J\big)^{\mathsf{T}} \qquad (9)$$

and

$$S^{(K)} = (E)_K\big((H_{\mathrm{GRS}})_K\big)^{\mathsf{T}} \,. \qquad (10)$$

Clearly, the syndrome array S that corresponds to E can be decomposed into

$$S = S^{(J)} + S^{(K)} \,,$$

and after multiplying the respective bivariate polynomials by the erasure-locator polynomial $\mathrm{M}(x) = \prod_{j\in K}(1-\alpha_j x)$, we get

$$\begin{aligned}
\sigma(y,x) &\equiv S(y,x)\,\mathrm{M}(x)\\
&\equiv S^{(J)}(y,x)\,\mathrm{M}(x) + S^{(K)}(y,x)\,\mathrm{M}(x) \pmod{x^{d-1}} \,.
\end{aligned} \qquad (11)$$

Now, recall (from the key equation) that the coefficients of $x^r, x^{r+1}, \ldots, x^{d-2}$ in $S^{(K)}(y,x)\,\mathrm{M}(x)$ are all zero, which means that the respective coefficients in $\sigma(y,x)$ are equal to those in $S^{(J)}(y,x)\,\mathrm{M}(x)$.

Let $\tilde{S}$ denote the $m \times (d-1-r)$ matrix

$$\tilde{S} = (\sigma_{h,i})_{h\in\langle m\rangle, i\in\langle r, d-1\rangle} \,.$$

It follows from (9)–(11) that

$$\tilde{S} = (E)_J\big((A_{\mathrm{M}}H_{\mathrm{GRS}})_J\big)^{\mathsf{T}} \,, \qquad (12)$$

where A_{M} is a $(d-1-r) \times (d-1)$ matrix over F whose first row consists of the $r+1$ coefficients of $\mathrm{M}(x)$ in decreasing order (padded with $d-2-r$ zero entries), and each subsequent row is obtained from its predecessor by a shift one position to the right (compare (12) with (3)). Hence, any column indexed by $j \in J$ in the $(d-1-r) \times t$ matrix $(A_{\mathrm{M}}H_{\mathrm{GRS}})_J$ takes the form

$$\alpha_j^r \cdot \mathrm{M}(\alpha_j^{-1}) \cdot \big(1 \ \alpha_j \ \alpha_j^2 \ \ldots \ \alpha_j^{d-2-r}\big)^{\mathsf{T}}$$

(where $\mathrm{M}(\alpha_j^{-1}) \neq 0$ since $K \cap J = \emptyset$). Hence, under the assumption that $t \le d-1-r$ (which, in fact, holds whenever $2t + r \le d + \mu - 2$) we get that $\mathsf{rank}\big((A_{\mathrm{M}}H_{\mathrm{GRS}})_J\big) = t$. It now follows from (12) that

$$\mathsf{colspan}\big((E)_J\big) = \mathsf{colspan}(\tilde{S}) \qquad (13)$$

(provided that $t \le d-1-r$); equivalently, for every $\boldsymbol{a} \in F^m$,

$$\boldsymbol{a}(E)_J = \boldsymbol{0} \quad \Longleftrightarrow \quad \boldsymbol{a}\tilde{S} = \boldsymbol{0} \,.$$

Input:
- Array Y of size $m \times n$ over F.
- Set K of size r of indexes of column erasures.

Steps:
1) Compute the $m \times (d-1)$ syndrome array

$$S = YH_{\mathrm{GRS}}^{\mathsf{T}} \,.$$

2) Compute the modified syndrome array to be the unique $m \times (d-1)$ matrix σ that satisfies the congruence:

$$\sigma(y,x) \equiv S(y,x)\,\mathrm{M}(x) \pmod{x^{d-1}} \,,$$

where

$$\mathrm{M}(x) = \prod_{j\in K}(1-\alpha_j x) \,.$$

Let μ be the rank of the $m \times (d-1-r)$ matrix $\tilde{S}$ formed by the columns of σ that are indexed by $\langle r, d-1\rangle$.

3) Using the Feng–Tzeng algorithm, compute a polynomial $\lambda(x)$ of (smallest) degree $\Delta \le (d+\mu-r)/2$ such that the following congruence is satisfied for some polynomial $\omega(y,x)$ with $\deg_x\omega(y,x) < r + \Delta$:

$$\sigma(y,x)\lambda(x) \equiv \omega(y,x) \pmod{x^{d-1}} \,.$$

If no such $\lambda(x)$ exists or the computed $\lambda(x)$ does not divide $\prod_{j\in\langle n\rangle}(1-\alpha_j x)$ then declare decoding failure and **Stop**.

4) Compute the $m \times n$ error array E by

$$E_j(y) = \begin{cases} \dfrac{-\alpha_j\cdot\omega(y,\alpha_j^{-1})}{\lambda'(\alpha_j^{-1})\cdot\mathrm{M}(\alpha_j^{-1})} & \text{if } \lambda(\alpha_j^{-1}) = 0\\[2ex] \dfrac{-\alpha_j\cdot\omega(y,\alpha_j^{-1})}{\lambda(\alpha_j^{-1})\cdot\mathrm{M}'(\alpha_j^{-1})} & \text{if } j \in K\\[2ex] 0 & \text{otherwise}\end{cases},$$

where $(\cdot)'$ denotes formal differentiation.

Output:
- Decoded array $Y - E$ of size $m \times n$.

Fig. 2. Decoding of interleaved GRS codes. (See Section II-B.)

In particular, $\mathsf{rank}\big((E)_J\big) = \mathsf{rank}(\tilde{S})$. In fact, every $m \times s$ sub-matrix of $\tilde{S}$ which consists of $s \ge t$ consecutive columns of $\tilde{S}$ has the same rank as $(E)_J$.

Given $\sigma(y,x) = \sum_{h\in\langle m\rangle}\sigma_h(x)y^h$ and the number of erasures r, we can use the Feng–Tzeng algorithm [9] to find efficiently a polynomial $\lambda(x) = \sum_{i=0}^{\Delta}\lambda_i x^i$ in $F[x]$ of smallest degree Δ such that (P1) is satisfied for some $\omega(y,x) = \sum_{h\in\langle m\rangle}\omega_h(x)y^h$ where $\deg_x\omega(y,x) < r + \Delta$. In other words, for every $h \in \langle m\rangle$, the sequence $(\sigma_{h,i})_{i=r}^{d-2}$ satisfies the linear recurrence

$$\sum_{i=0}^{\Delta}\lambda_i\sigma_{h,j-i} = 0 \,, \quad r + \Delta \le j \le d-2 \,.$$

Under the assumption that $2t + r \le d + \mu - 2$, we get from Lemma 3 that the polynomial $\lambda(x)$ found equals the error-locator polynomial $\Lambda(x)$ (up to a normalization by a constant). From the roots of $\lambda(x)$ one can then recover the set J.

The decoding algorithm is summarized in Figure 2. Next, we analyze its complexity. The syndrome computation step (Step 1 in the figure) can be carried out using $O(dmn)$ operations in F. Step 2 requires $O(drm) = O(d^2m)$ oper-

ations to compute $\sigma(y,x)$ and $O\big(dm \cdot \min(d,m)\big)$ operations to compute the rank μ. The application of the Feng–Tzeng algorithm in Step 3 requires $O(d^2m)$ operations, and, finally, Step 4 requires $O(dn)$ operations for the Chien search and $O(d^2m)$ operations for computing the nonzero columns of E. Overall, the decoding complexity amounts to $O(dmn)$ operations for syndrome computation, $O(dn)$ for the Chien search, and $O(d^2m)$ for the remaining steps.

We note that Lemma 3 and the decoding algorithm in Figure 2 apply also to the more general class of alternant codes over F, with d now standing for the designed minimum distance of the code. Specifically, we apply the lemma and the decoding algorithm to the underlying GRS code over the appropriate extension field of F: that GRS code has minimum distance d and contains the alternant code as a sub-field sub-code.

C. Application to Probabilistic Decoding

We next provide an application of the efficient decoding algorithm of Figure 2, for the following channel model: an $m \times n$ transmitted array over F is subject to at most a prescribed number t of block errors (and no erasures), such that the value (*i.e.*, contents) of the block error in each affected column is uniformly distributed over F^m, independently of the other block-error values. (Note that in this formulation of the channel, there is a probability of $1/q^m$ that an affected column will in fact be error-free; we have elected to define the channel this way in order to simplify the analysis.)

We consider the decoding problem given that the $m \times n$ transmitted array belongs to the code described in Section II-B, namely, an m-level interleaving of an $[n, k, d{=}n{-}k{+}1]$ GRS code over F. Let J be the index set of the columns that were affected (possibly by an error-free block error), where $|J| \le t$, and let μ be the rank of the error array E. The decoder of Figure 2 will fail to decode (or will decode incorrectly) *only* when the inequality in Lemma 3 is violated, namely, only when $\mu \le 2|J| - d + 1$. Under the assumed statistical model on the error arrays, it is easy to see that

$$\mathsf{Prob}\{\mathrm{rank}(E) \le \mu\} = \mathsf{Prob}\{\mathrm{rank}((E)_J) \le \mu\}$$
$$= q^{-(m-\mu)(|J|-\mu)} \cdot (1 + o(1))$$

(see [20, p. 699]), where $o(1)$ is an expression that goes to 0 as $q \to \infty$. Hence, when $m \ge d - 1$, the decoding failure probability of the algorithm in Figure 2 is bounded from above, up to a multiplicative factor $1 + o(1)$, by

$$q^{-(m+d-1-2|J|)(d-1-|J|)} \le q^{-(m+d-1-2t)(d-1-t)}.$$

For $m \ge d - 1$, this bound is (considerably) better than those given in [3] and [18], and is comparable to that in [27] when m is much larger than d.

III. MAIN CODE CONSTRUCTION

We come now to the main code construction of this paper, namely the code construction $\mathbb{C} = (\mathcal{C}, H_{\mathrm{in}})$ that was outlined in Section I. In particular, Section III-A gives all the details of the channel model and the code construction, Section III-B

presents the error and erasure correction capabilities of the code, and Section III-C discusses a variety of embodiments based on specific choices for the code $\mathcal{C}$ and the matrix H_{in}, and compares them with alternative code constructions. (Decoding algorithms for $\mathbb{C}$ will be discussed in Section IV.)

A. Channel Model and Code Definition

We consider the following channel model. An $m \times n$ array Γ over F is stored (or transmitted), and Γ is subject to the following error and erasure types (see also Table I and Figure 1):

(T1) *Block errors:* a subset of columns in Γ that are indexed by $\mathcal{J} \subseteq \langle n \rangle$ can be erroneous.
(T2) *Block erasures:* a subset of columns in Γ that are indexed by $\mathcal{K} \subseteq \langle n \rangle \setminus \mathcal{J}$ can be erased.
(T3) *Symbol errors:* a subset of entries in Γ that are indexed by $\mathcal{L} \subseteq \langle m \rangle \times (\langle n \rangle \setminus (\mathcal{K} \cup \mathcal{J}))$ can be erroneous.
(T4) *Symbol erasures:* a subset of entries in Γ that are indexed by $\mathcal{R} \subseteq \big(\langle m \rangle \times (\langle n \rangle \setminus \mathcal{K})\big) \setminus \mathcal{L}$ can be erased.

Let the $m \times n$ matrix $\mathcal{E}$ over F represent the alterations that happen to Γ over time (or during transmission). Then the read out (or received) message is given by the $m \times n$ matrix

$$\Upsilon = \Gamma + \mathcal{E}$$

over F. Note that our formalism treats erasures like errors, with the side information $\mathcal{K}$ and $\mathcal{R}$ telling us their location. (Of course, the sets $\mathcal{J}$ and $\mathcal{L}$ are not known to the decoder *a priori*.)

Write $\tau = |\mathcal{J}|$, $\rho = |\mathcal{K}|$, $\vartheta = |\mathcal{L}|$, and $\varrho = |\mathcal{R}|$. The total number of symbol errors (resulting from error types (T1) and (T3)) is at most $m\tau + \vartheta$ and the total number of symbol erasures (resulting from erasure types (T2) and (T4)) is at most $m\rho + \varrho$; hence, we should be able to correct all error and erasure types (T1)–(T4) (when occurring simultaneously) while using a code of length mn over F with minimum distance at least $m(2\tau + \rho) + 2\vartheta + \varrho + 1$. However, such a strategy does not take into account the fact that errors of type (T1)–(T2) are aligned across the m rows of the $m \times n$ array Γ. The next construction is designed to take advantage of such an alignment.

Definition 4 *Let $\mathcal{C}$ be a linear $[n, k, d]$ code over F, and let H_{in} be an $m \times (mn)$ matrix over F that satisfies the following two properties for some positive integer δ:*

(a) *Every subset of $\delta - 1$ columns in H_{in} is linearly independent (namely, H_{in} is a parity-check matrix of a linear code over F of length mn and minimum distance at least δ), and—*
(b) *writing*

$$H_{\mathrm{in}} = \big(\ H_0 \mid H_1 \mid \dots \mid H_{n-1}\ \big),$$

with $H_0, H_1, \dots, H_{n-1}$ being $m \times m$ sub-matrices of H_{in}, each H_j is invertible over F.

Given $\mathcal{C}$ and H_{in}, we define $\mathbb{C} = (\mathcal{C}, H_{\mathrm{in}})$ to be the linear $[mn, mk]$ code over F which consists of all $m \times n$ matrices

$$\Gamma = \big(\ \Gamma_0 \mid \Gamma_1 \mid \dots \mid \Gamma_{n-1}\ \big)$$

TABLE I
TYPES OF ERRORS AND ERASURES UNDER CONSIDERATION.

	error	erasure
block	**(T1)** column set $\mathcal{J}$ $\|\mathcal{J}\| = \tau$	**(T2)** column set $\mathcal{K}$ $\|\mathcal{K}\| = \rho$
symbol	**(T3)** location set $\mathcal{L}$ $\|\mathcal{L}\| = \vartheta$	**(T4)** location set $\mathcal{R}$ $\|\mathcal{R}\| = \varrho$

over F (where Γ_j stands for column j of Γ) such that each row in

$$Z = \left(\; H_0\Gamma_0 \mid H_1\Gamma_1 \mid \ldots \mid H_{n-1}\Gamma_{n-1} \; \right) \qquad (14)$$

is a codeword of $\mathcal{C}$. $\qquad\square$

One can view the code $\mathbb{C}$ as a (generalized) concatenated code, where the outer code is an m-level interleaving of $\mathcal{C}$, such that an $m \times n$ matrix

$$Z = \left(\; Z_0 \mid Z_1 \mid \ldots \mid Z_{n-1} \; \right)$$

over F is an outer codeword if and only if each row in Z belongs to $\mathcal{C}$. Each column in Z then undergoes encoding by an inner encoder of rate one, where the encoder of column j is given by the bijective mapping $Z_j \mapsto H_j^{-1}Z_j$.

B. Error Correction Capabilities

This subsection discusses what combinations of errors and erasures of the types (T1)–(T4) can be handled by the code $\mathbb{C} = (\mathcal{C}, H_{\mathrm{in}})$ that was specified in Definition 4.

Theorem 5 *There exists a decoder for the code $\mathbb{C}$ that correctly recovers the transmitted array in the presence of errors of types (T1)–(T4) (which may occur simultaneously), whenever τ $(= \|\mathcal{J}\|)$, ρ $(= \|\mathcal{K}\|)$, ϑ $(= \|\mathcal{L}\|)$, and ϱ $(= \|\mathcal{R}\|)$ satisfy*

$$2\tau + \rho \le d - 2 \,,$$
$$2\vartheta + \varrho \le \delta - 1 \,.$$

Proof: Using puncturing as in the proof of Theorem 2, it suffices to prove the theorem for the case where $\mathcal{K}$ is empty, *i.e.*, there are no erasure events of type (T2).

The proof is by contradiction. So, assume that $\mathcal{E}$ and $\hat{\mathcal{E}}$ are two distinct $m \times n$ matrices that correspond to error events of types (T1), (T3), and (T4), with the respective sets $\mathcal{J}$, $\hat{\mathcal{J}}$, $\mathcal{L}$, and $\hat{\mathcal{L}}$ satisfying

$$2\tau_{\max} \quad\;\; \le d - 2 \,,$$
$$2\vartheta_{\max} + \varrho \le \delta - 1 \,,$$

where

$$\tau_{\max} = \max\{\|\mathcal{J}\|, \|\hat{\mathcal{J}}\|\} \,,$$
$$\vartheta_{\max} = \max\{\|\mathcal{L}\|, \|\hat{\mathcal{L}}\|\} \,,$$

and $\varrho = \|\mathcal{R}\|$ (the symbol erasure set $\mathcal{R}$ is the same for both $\mathcal{E}$ and $\hat{\mathcal{E}}$). We will have reached a contradiction once we have shown that $\mathcal{E} - \hat{\mathcal{E}}$ is a codeword of $\mathbb{C}$ if only if $\mathcal{E} = \hat{\mathcal{E}}$.

Let Z be the $m \times n$ matrix

$$Z = \left(\; H_0(\mathcal{E}_0{-}\hat{\mathcal{E}}_0) \mid H_1(\mathcal{E}_1{-}\hat{\mathcal{E}}_1) \mid \ldots \mid H_{n-1}(\mathcal{E}_{n-1}{-}\hat{\mathcal{E}}_{n-1}) \; \right) \,.$$

Observe that since the matrices $H_0, H_1, \ldots, H_{n-1}$ are all invertible over F, we get that a column in Z is zero if and only if it is zero in $\mathcal{E} - \hat{\mathcal{E}}$; in particular, $Z = 0$ if and only if $\mathcal{E} = \hat{\mathcal{E}}$. Write

$$\mathcal{Q} = \mathsf{supp}(\mathcal{E} - \hat{\mathcal{E}}) \setminus (\mathcal{J} \cup \hat{\mathcal{J}}) \,,$$

namely, the set $\mathcal{Q}$ indexes the columns of $\mathcal{E} - \hat{\mathcal{E}}$ that contain errors of type (T3) and (possibly part of) the erasures of type (T4). For each $j \in \mathcal{Q}$, denote by w_j the number of nonzero entries in $\mathcal{E}_j - \hat{\mathcal{E}}_j$ (that is, $w_j = \|\mathsf{supp}((\mathcal{E}_j - \hat{\mathcal{E}}_j)^{\mathsf{T}})\|$). The total number of nonzero entries in $(\mathcal{E} - \hat{\mathcal{E}})_\mathcal{Q}$ satisfies:

$$\sum_{j \in \mathcal{Q}} w_j \le \|\mathcal{L} \cup \hat{\mathcal{L}} \cup \mathcal{R}\| \le 2\vartheta_{\max} + \varrho \le \delta - 1 \,.$$

Consider the respective columns in Z:

$$Z_j = H_j(\mathcal{E}_j - \hat{\mathcal{E}}_j) \,, \quad j \in \mathcal{Q} \,.$$

Each Z_j is a nontrivial linear combination of w_j columns of H_{in}, and, obviously, for distinct indexes j these combinations involve disjoint sets of columns of H_{in}. Recalling that every subset of $\sum_{j \in \mathcal{Q}} w_j$ $(\le \delta - 1)$ columns in H_{in} is linearly independent, we thus get that the set of columns of $(Z)_\mathcal{Q}$ is linearly independent, that is,

$$\mathsf{rank}(Z) \ge \mathsf{rank}((Z)_\mathcal{Q}) = \|\mathcal{Q}\| \,.$$

On the other hand,

$$\|\mathsf{supp}(Z)\| \le \|\mathcal{J}\| + \|\hat{\mathcal{J}}\| + \|\mathcal{Q}\| \le 2\tau_{\max} + \|\mathcal{Q}\| \le d - 2 + \|\mathcal{Q}\|$$

and, so,

$$\|\mathsf{supp}(Z)\| - \mathsf{rank}(Z) \le d - 2 \,.$$

Hence, we conclude from Lemma 1 that each row in Z belongs to $\mathcal{C}$ only if $Z = 0$. Equivalently, $\mathcal{E} - \hat{\mathcal{E}}$ belongs to $\mathbb{C}$ if only if $\mathcal{E} = \hat{\mathcal{E}}$, as promised. $\qquad\blacksquare$

Remark 6 We draw the attention of the reader to the condition on τ and ρ in Theorem 5, namely, that the expression $2\tau + \rho$ be at most $d - 2$, rather than the (more common) requirement that it be at most $d - 1$. It is this slightly stronger condition that, implicitly, provides the required redundancy for correcting the (additional) symbol errors and erasures. $\qquad\square$

Remark 7 Theorem 5 indirectly implies a dependence of the correction capability of errors of type (T3)–(T4) on the parameter m, which, in turn, is part of the specification of the errors of type (T3)–(T4). Specifically, the largest possible value for δ can be the minimum distance of any linear code of length mn and redundancy m over F. Of course, one may re-arrange the array by grouping together non-overlapping sets of s columns, for some integer $s > 0$, to form an $m' \times n'$ array where $m' = sm$ and $n' = n/s$ (assuming the latter ratio is an

integer). The block errors and the symbol errors will remain so also in this modified setting, except that the block errors will be more structured than just being phased with respect to the (new) parameter m'. If this additional structure is not taken into account, the code $\mathbb{C}$ is bound to be sub-optimal. $\square$

Remark 8 In contrast to the previous remark, if m is (much) larger than the redundancy needed from a linear code of length mn and minimum distance δ as in Theorem 5, we may partition each column in the original array into s new columns, thereby forming an $m' \times n'$ array, where $m' = m/s$ and $n' = sn$, such that m' is (just) the redundancy required from a linear code of length mn and minimum distance δ. Of course, this will increase the number of block errors by a factor of s, yet as long as n' is sufficiently small to allow us to use a maximum-distance separable (MDS) code for $\mathcal{C}$ (as will be the case in the examples in Section III-C), we will incur a gain in the redundancy: it will reduce from $(2\tau + \rho)m + m$ to $(2\tau + \rho)m + m'$. Thus, the code $\mathbb{C}$ is suitable for (the rather practical) scenarios where the number of symbol errors is relatively small compared to the block-error length m. $\square$

C. Examples

In this subsection, we consider various special choices for $\mathcal{C}$ and H_{in} and demonstrate the properties of the resulting code $\mathbb{C}$ (Examples 9 through 12). We also demonstrate the advantages of the code $\mathbb{C}$ over existing alternatives for handling errors of types (T1)–(T4) (Examples 13 through 16).

Example 9 We consider first the special case where $mn \le q+1$. Here, we can take $\mathcal{C}$ to be an MDS code over F and H_{in} to be a parity-check matrix of an MDS code over F. Under such circumstances we have $d = n - k + 1$ and $\delta = m + 1$, which means that it suffices that the sizes τ, ρ, ϑ, and ϱ satisfy

$$2\tau + \rho \le n - k - 1 ,$$
$$2\vartheta + \varrho \le m .$$

The redundancy of $\mathbb{C}$, being $m(n-k)$, is then the smallest possible for this correction capability: since the total number of symbol errors is $m\tau + \vartheta$ and the total number of symbol erasures is $m\rho + \varrho$, by the Reiger bound ([19], [23]) we need a redundancy of at least $m(2\tau + \rho) + 2\vartheta + \varrho$ symbols over F in order to be able to correct all error types (T1)–(T4). Admittedly, the same performance of correction capability versus redundancy can be achieved also by a single linear $[mn, mk]$ MDS code C over F (which exists under the assumption that $mn \le q+1$). However, as pointed out earlier, the use of such a code C does not take into account the alignment of error types (T1) and (T2) across the rows of the received $m \times n$ array. It is this alignment that allows $\mathbb{C}$ to achieve the same correction capability using a code $\mathcal{C}$, which is m times shorter than C. While we still need for $\mathbb{C}$ the parity-check matrix H_{in} of an MDS code of length mn, the redundancy of the latter code needs to be only m, rather than $m(n-k)$.

To demonstrate the savings that $\mathbb{C}$ may offer compared to C, consider the simple problem of verifying whether a given $m \times n$ array Γ belongs to the code (namely, *detecting* whether errors have occurred). When using C, we will regard Γ as a vector of length mn over F and the checking will be carried out through multiplication by an $(m(n-k)) \times (mn)$ (systematic) parity-check matrix of C, thereby requiring up to $2m^2k(n-k)$ operations (namely, additions and multiplications) in F. In contrast, when using $\mathbb{C}$, we will first compute the array Z as in (14) while requiring less than $2m^2(n-1)$ operations in F (one of the matrices H_j can be assumed to be the identity matrix); then we will compute the syndrome of each row of Z, for which we will need up to $2mk(n-k)$ operations in F. $\square$

Example 10 Suppose that $mn \le q-1$ and select $\mathcal{C}$ to be an $[n, k, d=n-k+1]$ GRS code $\mathcal{C}_{\text{GRS}}$ over F as in Section II-B. For every $j \in \langle n \rangle$, let

$$H_j = \left(\beta_{\kappa,j}^h \right)_{h \in \langle m \rangle, \kappa \in \langle m \rangle} ,$$

such that the elements $\beta_{\kappa,j}$ are distinct and nonzero in F for all $\kappa \in \langle m \rangle$ and $j \in \langle n \rangle$; the respective matrix $H_{\text{in}} = (H_j)_{j \in \langle n \rangle}$ is then a parity-check matrix of an $[mn, m(n-1), m+1]$ GRS code over F. Given any $m \times n$ matrix $\Gamma = (\Gamma_{\kappa,j})_{\kappa \in \langle m \rangle, j \in \langle n \rangle}$ over F, the entries of $H_j \Gamma_j$ are given by

$$(H_j \Gamma_j)_h = \sum_{\kappa \in \langle m \rangle} \Gamma_{\kappa,j} \beta_{\kappa,j}^h , \quad h \in \langle m \rangle .$$

Hence, Γ is in $\mathbb{C} = (\mathcal{C}, H_{\text{in}})$ if and only if

$$\sum_{\kappa \in \langle m \rangle} \sum_{j \in \langle n \rangle} \Gamma_{\kappa,j} \alpha_j^i \beta_{\kappa,j}^h = 0 , \quad h \in \langle m \rangle , \quad i \in \langle d - 1 \rangle . \tag{15}$$

(Note that if $\beta_{\kappa,j}$ depended only on κ, then $\mathbb{C}$ could be seen as a two-dimensional shortening of a two-dimensional cyclic code; see, for example [26].) $\square$

Example 11 We show that sometimes the construction $\mathbb{C}$ in Example 10 is an MDS code. Assume therein that m divides $q - 1$ and that for every $j \in \langle n \rangle$, the multiplicative order of α_j divides $(q-1)/m$ (thus, each α_j has m distinct mth roots in F). For every $j \in \langle n \rangle$, select $\beta_{0,j}, \beta_{1,j}, \ldots, \beta_{m-1,j}$ to be the distinct roots of α_j in F. Then $\Gamma \in \mathbb{C}$ if and only if

$$\sum_{\kappa \in \langle m \rangle} \sum_{j \in \langle n \rangle} \Gamma_{\kappa,j} \beta_{\kappa,j}^{h+mi} = 0 , \quad h \in \langle m \rangle , \quad i \in \langle d - 1 \rangle .$$

The latter condition, in turn, is equivalent to Γ being a codeword of a GRS code of length mn and redundancy $(d-1)m$ over F. $\square$

In contrast, the following example shows that sometimes $\mathbb{C} = (\mathcal{C}, H_{\text{in}})$ is not an MDS code, even when $\mathcal{C}$ is MDS and H_{in} is a parity-check matrix of an MDS code.

Example 12 Suppose that q is a power of 4 and take $m = 3$ and $n = (q+2)/3$. Select H_{in} to be a parity-check matrix of a $[q+2, q-1, 4]$ triply-extended GRS code over F [20, p. 326] and $\mathcal{C}$ to be any linear $[n, k]$ code over F. Thus, $\mathbb{C}$ is a linear $[q+2, 3k]$ code over F. It follows from the already-proved

range of the MDS conjecture that $\mathbb{C}$, being longer than $q+1$, cannot be MDS when, say, $2 \le k \le 1 + \frac{1}{6}\sqrt{q}$ [28]. $\square$

In fact, Example 12 shows that there are choices of F, n, m, τ, and ϑ for which the construction $\mathbb{C} = (\mathcal{C}, H_{\text{in}})$ can be realized to correct any τ block errors and any ϑ symbol errors, while, on the other hand, there are no codes over F of the same length and size as $\mathbb{C}$ that can correct any $2\tau m + \vartheta$ symbol errors. Examples 13 and 14 below present a larger range of parameters where this may happen.

In Examples 13–16, we make a running assumption that $n \le q$, in which case $\mathcal{C}$ can be taken as an MDS code, such as a (possibly extended) GRS code. For the sake of simplicity, we will consider in these examples only the block–symbol error-only case, *i.e.*, no erasures are present.

Example 13 Given positive τ, ϑ, n, and $F = \text{GF}(q)$ (such that $2\tau + 2 \le n \le q$), we take H_{in} to be a parity-check matrix of a (possibly extended) shortened BCH code of length mn over F, where m is determined by ϑ, n, and q to satisfy the equality

$$m = 1 + \left\lceil \frac{q-1}{q} \cdot (2\vartheta - 1) \right\rceil \cdot \left\lceil \log_q(mn) \right\rceil$$

(so mn may be larger than q; see [24, p. 260]). The code $\mathcal{C}$ is taken as a (possibly extended) $[n, k, d]$ GRS code over F where $d = 2\tau + 2$. The overall redundancy of $\mathbb{C} = (\mathcal{C}, H_{\text{in}})$ is then

$$(2\tau + 1)m = 2\tau m + 1 + \left\lceil \frac{q-1}{q} \cdot (2\vartheta - 1) \right\rceil \cdot \left\lceil \log_q(mn) \right\rceil . \tag{16}$$

The first term, $2\tau m$, on the right-hand side of (16) is the smallest redundancy possible if one is to correct any τ block errors of length m. The remaining term therein is the redundancy (or an upper bound thereof) of a BCH code that corrects any ϑ symbol errors over F. In comparison, a shortened BCH code of length mn over F that corrects any $\tau m + \vartheta$ symbol errors may have redundancy as large as

$$1 + \left\lceil \frac{q-1}{q} \cdot \big(2(\tau m + \vartheta) - 1\big) \right\rceil \cdot \left\lceil \log_q(mn) \right\rceil .$$

It can be verified that the latter expression is larger than (16) when $mn > q \ge 4$. $\square$

Example 14 We repeat the previous example, except that we replace the shortened BCH codes with codes that attain the Gilbert–Varshamov bound. Denote by $\text{red}_q(N, D)$ the redundancy of such a linear code of length N and minimum distance D over F. Then m is determined by ϑ, n, and q through

$$m = \text{red}_q(mn, 2\vartheta + 1) .$$

Taking again $\mathcal{C}$ to be an $[n, k, d]$ GRS code over F, the code $(\mathcal{C}, H_{\text{in}})$ has redundancy

$$(2\tau + 1)m = 2\tau m + \text{red}_q(mn, 2\vartheta + 1) ,$$

while a linear code that attains the Gilbert–Varshamov bound and corrects any $\tau m + \vartheta$ symbol errors has redundancy

$$\text{red}_q\big(mn, 2(\tau m + \vartheta) + 1\big) .$$

Thus, we are interested in the range in which the following inequality holds:

$$\text{red}_q\big(mn, 2(\tau m + \vartheta) + 1\big) \ge 2\tau m + \text{red}_q(mn, 2\vartheta + 1) . \tag{17}$$

As shown in Appendix A, the inequality (17) indeed holds whenever

$$2(\tau m + \vartheta - 1) \le mn \cdot \frac{q-1}{2q-1} \tag{18}$$

which, in turn, holds whenever

$$2\tau + 1 \le n \cdot \frac{q-1}{2q-1}$$

(recall that $2\vartheta \le m$). $\square$

Example 15 We compare the performance of $\mathbb{C}$ with that of a concatenated code C constructed from a linear $[m, \mathsf{k}, \mathsf{d}]$ inner code over F and a linear $[n, \mathsf{K}, \mathsf{D}]$ outer code over $\text{GF}(q^{\mathsf{k}})$ (where $n \le q$). By the Singleton bound, we can bound the redundancy of C from below by

$$mn - \mathsf{kK} \ge (\mathsf{D}-1)m + (\mathsf{d}-1)n - (\mathsf{D}-1)(\mathsf{d}-1)$$
$$= (\mathsf{D}-1)(m+1) - n + \mathsf{d}(n-\mathsf{D}+1) . \tag{19}$$

As we mentioned already in Section I-B, any error pattern of up to τ block errors and up to ϑ symbol errors can be correctly decoded, whenever

$$2\vartheta + 1 \le \mathsf{d}(\mathsf{D} - 2\tau) . \tag{20}$$

Since in our setting the values τ and ϑ are prescribed, we can minimize (19) over d and D, subject to the inequality (20). Specifically, we define $\Delta = \mathsf{D} - 2\tau$ and, from (20), we can express d in terms of Δ as $\mathsf{d} = \lceil (2\vartheta + 1)/\Delta \rceil$, in which case (19) becomes

$$\Delta(m+1) + \left\lceil \frac{2\vartheta + 1}{\Delta} \right\rceil (n - 2\tau + 1 - \Delta)$$
$$+ (2\tau - 1)(m+1) - n \tag{21}$$
$$\ge \Delta(m+1) + \frac{(2\vartheta + 1)(n - 2\tau + 1)}{\Delta}$$
$$- (2\vartheta + 1) - (n - 2\tau + 1) + (2\tau - 1)m . \tag{22}$$

The minimum of (22) over (a real) Δ is attained at

$$\Delta_{\min} = \sqrt{\frac{(2\vartheta + 1)(n - 2\tau + 1)}{m + 1}} .$$

Yet we need to take into account that both d and Δ are positive integers.

Case 1: $n - 2\tau < \frac{m+1}{2\vartheta+1}$. In this range (of very small n) we have $\Delta_{\min} \le 1$ and, so, we take $\Delta = 1$; namely, for this range, it is best to let the inner code handle (exclusively) the symbol errors, and then the outer code is left to correct the block errors. For this range, the expression (21) becomes

$$2\tau m + 2\vartheta(n - 2\tau) ,$$

which is never larger than the smallest possible redundancy, $(2\tau+1)m$, of $\mathbb{C}$. Notice, however, that when H_{in} in $\mathbb{C}$ can be taken as a parity-check matrix of an MDS code, then $m = 2\vartheta$ and therefore this range is empty.

Case 2: $n - 2\tau \geq (m+1)\cdot(2\vartheta+1)$. In this range we have $\Delta_{\min} > 2\vartheta+1$ and, so, the expression $\lceil\cdot\rceil$ (for d) in (21) equals 1; namely, for this range, it is best to regard the symbol errors as block errors. Therefore, we take $\Delta = 2\vartheta+1$ and the expression (21) becomes

$$2(\tau+\vartheta)m\,,$$

which is larger than the redundancy of $\mathbb{C}$ whenever $\vartheta > 0$ (assuming that $\mathcal{C}$ in $\mathbb{C}$ is an MDS code).

Case 3: $\frac{m+1}{2\vartheta+1} \leq n - 2\tau < (m+1)\cdot(2\vartheta+1)$. In this range, we plug the expression for $\Delta_{\min}$ into (22), resulting in the following lower bound expression on the redundancy of C:

$$2\sqrt{(2\vartheta+1)(n-2\tau+1)(m+1)} - (2\vartheta+1)$$
$$- (n-2\tau+1) + (2\tau-1)m\,.$$

This lower bound is greater than the redundancy of $\mathbb{C}$ whenever

$$4(2\vartheta+1)(n-2\tau+1)(m+1)$$
$$> \Big(2m + (2\vartheta+1) + (n-2\tau+1)\Big)^2\,. \quad (23)$$

Since the left-hand side of (23) is a cubic expression while the right-hand side is only quadratic, the inequality (23) is expected to hold for most parameters of interest. $\square$

Example 16 In many cases, the redundancy of $\mathbb{C}$ is smaller even than that of the generalized concatenated (GC) code construction defined through conditions (G1)–(G2) in Section I-B. Referring to the notation therein, we first note that if $\mathsf{D}_v > 2\tau+1$, then the contribution of condition (G2) alone to the redundancy is already at least $(\mathsf{D}_v-1)m \geq (2\tau+1)m$. Hence, we assume that $\mathsf{D}_v = 2\tau+1$, in which case, from (1), we get that $\mathsf{r}_v \geq \mathsf{d}_v - 1 \geq 2\vartheta$. Thus, condition (G2) induces a redundancy of at least $2\tau m$, and condition (G1) adds a redundancy of at least

$$\sum_{i=1}^{v} (\mathsf{r}_i - \mathsf{r}_{i-1})(\mathsf{D}_{i-1} - 1 - 2\tau)\,. \quad (24)$$

In Appendix B, we show that the expression (24) is bounded from below by

$$(2\vartheta+1)\ln\vartheta + 2\gamma\cdot\vartheta + O(1)\,, \quad (25)$$

where γ is Euler's constant (approximately 0.5772) [10, p. 264]. Taking $\mathcal{C}$ in $\mathbb{C}$ as an MDS code, the redundancy of $\mathbb{C}$ is then smaller than that of GC codes (with the same correction capabilities) whenever (25) exceeds m. $\square$

One may speculate whether Theorem 5 holds for the following more general definition of $\mathbb{C}$: instead of requiring that each row of Z in (14) be a codeword of $\mathcal{C}$, require that Z belong to a linear $[n, k, d]$ code over $\text{GF}(q^m)$, where each column of Z is regarded as an element of the latter field with respect to some basis of that field over F. It turns out that Theorem 5

does *not* hold for this more general setting, as shown by the counterexample that is discussed in the following example.

Example 17 Let $d = 2$ (*i.e.*, $\tau = \rho = 0$). Assume that $n \leq q^m - 1$ and let $C_0, C_1, C_2, \ldots, C_{n-1}$ be any n distinct powers of an $m\times m$ companion matrix of some irreducible polynomial of degree m over F [20, p. 106], where $C_0 = I$ (the $m \times m$ identity matrix). Then

$$\left\{ Z = (\ Z_0\ |\ Z_1\ |\ \ldots\ |\ Z_{n-1}\ )\ :\ \sum_{j\in\langle n\rangle} C_j Z_j = \mathbf{0} \right\}$$

is a linear $[n, n-1, 2]$ code over $\text{GF}(q^m)$ (with the columns of each Z being the codeword coordinates). The respective code $\mathbb{C}$ would then be written as

$$\left\{ \Gamma = (\ \Gamma_0\ |\ \Gamma_1\ |\ \ldots\ |\ \Gamma_{n-1}\ )\ :\ \sum_{j\in\langle n\rangle} C_j H_j \Gamma_j = \mathbf{0} \right\}\,.$$

If $\delta \geq 3$ then the $2m$ columns of H_0 and H_1 are all nonzero and distinct. Therefore, we can select $C_1 \neq I$ so that the first column (say) in $C_1 H_1$ equals the first column (say) in $H_0 = C_0 H_0$. Yet this means that there exists a nonzero $\Gamma \in \mathbb{C}$ that contains only two nonzero entries. Therefore, $\mathbb{C}$ cannot have a decoder that corrects all one-symbol error patterns. $\square$

IV. DECODING ALGORITHMS FOR THE MAIN CODE CONSTRUCTION

We now discuss a variety of decoders for the code $\mathbb{C} = (\mathcal{C}, H_{\text{in}})$ that was specified in Definition 4, for the case where the constituent code $\mathcal{C}$ is a GRS code. Section IV-A presents a polynomial-time decoding algorithm for error and erasures of types (T1)–(T4) as far as they are correctable as guaranteed by Theorem 5, and provided that the code parameters n, d, δ, ρ, and ϱ satisfy a certain inequality (see Theorem 19 below). Then, we consider the construction $\mathbb{C}$ as in Example 10 (where $\mathcal{C}$ is a GRS code and H_{in} is a parity-check matrix of a GRS code) and present some more specialized decoders for this construction. Namely, Section IV-B discusses a decoder that handles errors and erasures of types (T1), (T2), (T4), but not of type (T3), and Section IV-C introduces a decoder that handles errors and erasures of types (T1), (T2), (T4), and some combinations of errors of type (T3). As of yet, we do not have an efficient decoder that corrects all error patterns that satisfy the conditions of Theorem 5.

Assume that m, τ, ρ, ϑ, and ϱ scale linearly with n. If $\mathbb{C}$ is replaced by a GRS code (if such a code exists) then the decoding complexity scales linearly with $(n^2)^2 = n^4$. One of the main purposes of defining the code $\mathbb{C} = (\mathcal{C}, H_{\text{in}})$ and formulating the decoders in Sections IV-B and IV-C is that their decoding complexity scales linearly with n^3.

A. Polynomial-Time Decoding Algorithm

Example 11 demonstrates one particular instance of the construction $\mathbb{C} = (\mathcal{C}, H_{\text{in}})$ for which the decoder guaranteed by Theorem 5 has an efficient implementation (simply because in this case the code $\mathbb{C}$ is a GRS code). In this section, we exhibit a much wider range of instances for which decoding can be carried out in polynomial-time complexity.

Specifically, we consider the case where the code $\mathcal{C}$ is a GRS code over F (and, so, $n \le q - 1$), and H_{in} is an *arbitrary* $m \times (mn)$ matrix over F that satisfies the two properties (a)–(b) in Definition 4. The columns of $m \times n$ arrays will be regarded as elements of the extension field $\mathrm{GF}(q^m)$ (according to some basis of $\mathrm{GF}(q^m)$ over F). When doing so, the matrix Z in (14) can be seen as a codeword of a GRS code $\mathcal{C}'$ over $\mathrm{GF}(q^m)$, where $\mathcal{C}'$ has the same code locators $(\alpha_j)_{j \in \langle n \rangle}$ as $\mathcal{C}$.

Let $\Gamma \in \mathbb{C}$ be the transmitted $m \times n$ array and let Υ be the received $m \times n$ array, possibly corrupted by τ errors of type (T1) and ϑ errors of type (T3), where $\tau \le (d/2) - 1$ and $\vartheta \le (\delta - 1)/2$. We first compute an $m \times n$ array

$$ Y = \left(\; H_0 \Upsilon_0 \; \big| \; H_1 \Upsilon_1 \; \big| \; \ldots \; \big| \; H_{n-1} \Upsilon_{n-1} \; \right) \; , $$

where Y contains $\tau + \vartheta \le (d + \delta - 3)/2$ erroneous columns. Regarding now Y as a corrupted version of a codeword of $\mathcal{C}'$, we can apply a *list decoder* for $\mathcal{C}'$ to Y. Such a decoder returns a list of up to a prescribed number L of codewords of $\mathcal{C}'$, and the returned list is guaranteed to contain the correct codeword, provided that the number of erroneous columns in Y does not exceed the *decoding radius* of $\mathcal{C}'$. In our decoding, we will use the polynomial-time list decoder due to Guruswami and Sudan [11] (for variation of the algorithm that reduce its complexity, see [16] and [30]). For any GRS code of length n and minimum distance d (over any field) and for any prescribed list size L, their decoder will return the correct codeword as long as the number of errors does not exceed $\lceil n \Theta_L(d/n) \rceil - 1$, where $\Theta_L(d/n)$ is the maximum over $s \in \{1, 2, \ldots, L\}$ of the following expression:

$$ \Theta_{L,s}(d/n) = 1 - \frac{s+1}{2(L+1)} - \frac{L}{2s}\left(1 - \frac{d}{n} \right) $$

(see [24, Chapter 9.5]). Thus, if L is such that

$$ n\Theta_L(d/n) \ge (d + \delta - 1)/2 \; , \qquad (26) $$

then the returned list will contain the correct codeword

$$ Z = \left(\; H_0 \Gamma_0 \; \big| \; H_1 \Gamma_1 \; \big| \; \ldots \; \big| \; H_{n-1} \Gamma_{n-1} \; \right) $$

of $\mathcal{C}'$. For each array Z' in the list we can compute the respective array in $\mathbb{C}$,

$$ \Gamma' = \left(\; H_0^{-1} Z_0' \; \big| \; H_1^{-1} Z_1' \; \big| \; \ldots \; \big| \; H_{n-1}^{-1} Z_{n-1}' \; \right) \; , $$

and it follows from the proof Theorem 5 that only one such computed array Γ'—namely, the transmitted array Γ—can correspond to an error pattern of up to $(d/2) - 1$ block errors and up to $(\delta - 1)/2$ symbol errors. Finding that array can be done simply by checking each computed Z' against the received array Υ.

Remark 18 While the decoding scheme that we have just outlined makes essential use of an efficient list decoder for the m-level interleaving of $\mathcal{C}$, nothing is assumed about H_{in} beyond properties (a)–(b) in Definition 4. In particular, nothing is assumed about the decoding complexity of the code over F that is defined by the parity-check matrix H_{in}. $\qquad \square$

Our decoding scheme can be generalized to handle also erasure of types (T2) and (T4) by applying a list decoder for the GRS code obtained by puncturing $\mathcal{C}'$ on the columns that are affected by erasures: this translates into replacing d by $d - \rho - \varrho$ (assuming that the latter value is positive).

The next theorem characterizes a range of parameters for which (26) holds for some polynomially-large list size L (and, thus, $\mathbb{C}$ can be decoded in polynomial time).

Theorem 19 *For $\mathbb{C} = (\mathcal{C}, H_{\mathrm{in}})$ such that $\mathcal{C}$ is a GRS code over F, the decoder guaranteed by Theorem 5 can be implemented by a polynomial-time algorithm, whenever*

$$ d - \rho - \varrho \ge 2\sqrt{\delta(n - \rho - \varrho)} - \delta \qquad (27) $$

(or, simply, whenever $d \ge 2\sqrt{\delta n} - \delta$ in case $\rho = \varrho = 0$).

Proof: We will assume in the proof that $\rho = \varrho = 0$; the general case follows by observing that any puncturing of $\mathcal{C}'$ on $\rho + \varrho$ positions results in a GRS code of length $n - \rho - \varrho$ and minimum distance $d - \rho - \varrho$.

Our proof will be complete once we identify a polynomially-large L for which (26) holds. We take L to be such that

$$ \frac{d}{n} \ge 2\sqrt{\frac{\delta - 1}{n}} - \frac{\delta - 1}{n} + \frac{2}{L+1} \; . \qquad (28) $$

It readily follows from (27) that there exists such an L which is at most quadratic in n.

Define s to be

$$ s = L - \left\lceil L \cdot \sqrt{\frac{\delta - 1}{n}} \right\rceil \; . $$

From (28) we have

$$ \frac{d}{n} \ge \frac{s}{L-s} \cdot \frac{\delta - 1}{n} + \frac{L + 1 - s}{L + 1} \; , $$

which can also be rewritten as

$$ \frac{d}{n} \ge \frac{2s}{L-s}\left(\frac{\delta - 1}{2n} + \frac{L}{2s} + \frac{s+1}{2(L+1)} - 1 \right) \; . $$

Multiplying both sides by $(L-s)/(2s)$ and rearranging terms yields

$$ 1 - \frac{s+1}{2(L+1)} - \frac{L}{2s}\left(1 - \frac{d}{n} \right) \ge \frac{d}{2n} + \frac{\delta - 1}{2n} \; , $$

which is equivalent to

$$ \Theta_{L,s}(d/n) \ge \frac{d + \delta - 1}{2n} \; . $$

This, in turn, implies (26). $\qquad \blacksquare$

B. Decoding of Errors and Erasures of Type (T1), (T2), (T4), but not of Type (T3)

In this section, we present an efficient decoder for the code $\mathbb{C}$ when constructed as in Example 10, for the special case where $\vartheta = |\mathcal{L}| = 0$ (no errors of type (T3)).

An $m \times n$ matrix Γ is transmitted and an $m \times n$ matrix

$$ \Upsilon = \Gamma + \mathcal{E} $$

is received, where $\mathcal{E} = (\varepsilon_{\kappa,j})_{\kappa \in \langle m \rangle, j \in \langle n \rangle}$ is an $m \times n$ error matrix, with $\mathcal{J}$ $(\subseteq \langle n \rangle)$ (respectively, $\mathcal{K}$ $(\subseteq \langle n \rangle)$) indexing the columns in which block errors (respectively, block erasures) have occurred, and $\mathcal{R}$ $(\subseteq \langle m \rangle \times \langle n \rangle)$ is a nonempty set of positions where symbol erasures have occurred. We assume that d, τ $(= |\mathcal{J}|)$, and ρ $(= |\mathcal{K}|)$ satisfy

$$2\tau + \rho \leq d - 2 \qquad (29)$$

and that ϱ $(= |\mathcal{R}|)$ satisfies

$$0 < \varrho \leq m .$$

Define

$$Y = \left(\begin{array}{c|c|c|c} H_0 \Upsilon_0 & H_1 \Upsilon_1 & \ldots & H_{n-1} \Upsilon_{n-1} \end{array} \right)$$

and

$$E = (e_{h,j})_{h \in \langle m \rangle, j \in \langle n \rangle}$$
$$= \left(\begin{array}{c|c|c|c} H_0 \mathcal{E}_0 & H_1 \mathcal{E}_1 & \ldots & H_{n-1} \mathcal{E}_{n-1} \end{array} \right) .$$

Clearly,

$$Y = Z + E ,$$

where Z is given by (14). In particular, every row in Z is a codeword of $\mathcal{C}_{\mathrm{GRS}}$.

Next, write $\mathcal{R} = \{(\kappa_\ell, j_\ell)\}_{\ell \in \langle \varrho \rangle}$. Fix some $\ell \in \langle \varrho \rangle$ and define the following ϱ univariate polynomial (of degree $\varrho - 1$)

$$\mathrm{B}^{(\ell)}(y) = \sum_{i \in \langle \varrho \rangle} \mathrm{B}_i^{(\ell)} y^i \qquad (30)$$

$$= \prod_{(\kappa,j) \in \mathcal{R} \setminus \{(\kappa_\ell, j_\ell)\}} \frac{1 - \beta_{\kappa,j} y}{1 - \beta_{\kappa,j} \beta_{\kappa_\ell, j_\ell}^{-1}} , \quad \ell \in \langle \varrho \rangle , \quad (31)$$

and let $e^{(\ell)} = (e_j^{(\ell)})_{j \in \langle n \rangle}$ denote row $\varrho - 1$ of the $(m + \varrho - 1) \times n$ matrix $\mathrm{B}^{(\ell)}(y) E(y,x)$. We have

$$\mathsf{supp}(e^{(\ell)}) \subseteq \mathcal{J} \cup \mathcal{K} \cup \{j_\ell\} , \quad \ell \in \langle \varrho \rangle .$$

Indeed, the contribution of a symbol erasure at position (κ, j) in $\mathcal{E}$ to the column $E_j(y)$ of $E(y,x)$ is an additive term of the form

$$\varepsilon_{\kappa,j} \cdot \mathrm{T}_m(y; \beta_{\kappa,j}) = \varepsilon_{\kappa,j} \cdot \frac{1 - (\beta_{\kappa,j} y)^m}{1 - \beta_{\kappa,j} y}$$

(where we recall the definition of $\mathrm{T}_m(\cdot; \cdot)$ from Section I-C); so, if $(\kappa, j) \neq (\kappa_\ell, j_\ell)$ then the product

$$\mathrm{B}^{(\ell)}(y) \cdot \varepsilon_{\kappa,j} \cdot \frac{1 - (\beta_{\kappa,j} y)^m}{1 - \beta_{\kappa,j} y} = \varepsilon_{\kappa,j} \cdot \frac{\mathrm{B}^{(\ell)}(y)}{1 - \beta_{\kappa,j} y} \cdot (1 - (\beta_{\kappa,j} y)^m)$$

is a polynomial in which the powers $y^{\varrho-1}, y^\varrho, \ldots, y^{m-1}$ have zero coefficients.

Now, every row in the $(m + \varrho - 1) \times n$ array $Z^{(\ell)}(y, x) = \mathrm{B}^{(\ell)}(y) Z(y, x)$ is a codeword of $\mathcal{C}_{\mathrm{GRS}}$. Therefore, by applying a decoder for $\mathcal{C}_{\mathrm{GRS}}$ to row $\varrho - 1$ of $Z^{(\ell)}$ with $\rho + 1$ erasures indexed by $\mathcal{K} \cup \{j_\ell\}$, we should be able to decode the vector $e^{(\ell)}$, based on our assumption (29).

It follows from the definition of $e^{(\ell)}$ that for every $j \in \langle n \rangle$,

$$\begin{pmatrix} e_j^{(0)} \\ e_j^{(1)} \\ \vdots \\ e_j^{(\varrho-1)} \end{pmatrix} = \begin{pmatrix} \mathrm{B}_0^{(0)} & \mathrm{B}_1^{(0)} & \cdots & \mathrm{B}_{\varrho-1}^{(0)} \\ \mathrm{B}_0^{(1)} & \mathrm{B}_1^{(1)} & \cdots & \mathrm{B}_{\varrho-1}^{(1)} \\ \vdots & \vdots & \vdots & \vdots \\ \mathrm{B}_0^{(\varrho-1)} & \mathrm{B}_1^{(\varrho-1)} & \cdots & \mathrm{B}_{\varrho-1}^{(\varrho-1)} \end{pmatrix} \begin{pmatrix} e_{\varrho-1,j} \\ e_{\varrho-2,j} \\ \vdots \\ e_{0,j} \end{pmatrix} .$$

$$(32)$$

In particular,

$$e_{j_\ell}^{(\ell)} = \sum_{i \in \langle \varrho \rangle} \mathrm{B}_i^{(\ell)} \sum_{\kappa : (\kappa, j_\ell) \in \mathcal{R}} \varepsilon_{\kappa, j_\ell} \beta_{\kappa, j_\ell}^{\varrho - 1 - i}$$

$$= \sum_{\kappa : (\kappa, j_\ell) \in \mathcal{R}} \varepsilon_{\kappa, j_\ell} \beta_{\kappa, j_\ell}^{\varrho - 1} \mathrm{B}^{(\ell)}(\beta_{\kappa, j_\ell}^{-1})$$

$$= \varepsilon_{\kappa_\ell, j_\ell} \beta_{\kappa, j_\ell}^{\varrho - 1} .$$

Because $\ell \in \langle \varrho \rangle$ was arbitrary, we are able to recover the error values in $\mathcal{E}$ at the positions $\mathcal{R}$. Namely,

$$\varepsilon_{\kappa_\ell, j_\ell} = e_{j_\ell}^{(\ell)} \beta_{\kappa_\ell, j_\ell}^{1 - \varrho} , \quad \ell \in \langle \varrho \rangle .$$

This, in turn, allows us to eliminate the symbol erasures from E.

Figure 3 summarizes the decoding algorithm of a combination of errors of type (T1), (T2), and (T4). The complexity of Step 1 is $O((d + m)mn)$ operations in F (see the discussion that precedes Example 10). Step 2 requires $O(d\rho\varrho)$ operations. Each iteration in Step 3 requires $O(d\varrho)$ operations (for Step 3a), $O(d^2)$ operations (for Step 3b), and $O(dm)$ operations (for Step 3c), totaling to $O(d(d + m)\varrho)$ for Step 3. Step 4 requires $O(d^2 m)$ operations to compute the error-locator and error-evaluator polynomials, and additional $O(dn)$ for the Chien search. Finally, Step 5 requires $O(dm^2)$ operations. To summarize, the decoding complexity amounts to $O((d + m)mn)$ operations for syndrome computation, $O(dn)$ for the Chien search, and $O(d(d + m)m)$ for the remaining steps.

C. Decoding of Errors and Erasures of Type (T1), (T2), (T4), and with Restrictions on Errors of Type (T3)

In this section, we consider the decoding of $\mathbb{C}$ when constructed as in Example 10, under certain assumptions on the set $\mathcal{L}$, namely, under some restrictions on the symbol error positions (errors of type (T3)). These restrictions always hold when $|\mathcal{L}| \leq 3$ and d is sufficiently large.

Specifically, we consider the case where each column, except possibly for one column, contains at most one symbol error. The general strategy will be to locate the positions of these errors, thereby reducing to the case considered in Section IV-B. We use the same notation as in that section, except that the set $\mathcal{L}$ is not necessarily empty and that (for reasons of simplicity) the set $\mathcal{R}$ is empty. As in Section IV-B, the number τ of block errors and the number ρ of block erasures satisfy $2\tau + \rho \leq d - 2$.

When $\vartheta = |\mathcal{L}| > 0$, we write $\mathcal{L} = \{(\kappa_\ell, j_\ell)\}_{\ell \in \langle \vartheta \rangle}$, and assume that that there exists a $w \in \langle \vartheta \rangle$ such that the values

Input:

- Array Υ of size $m \times n$ over F.
- Set $\mathcal{K}$ of indexes of column erasures.
- Set $\mathcal{R} = \{(\kappa_\ell, j_\ell)\}_{\ell \in \langle \varrho \rangle}$ of positions of symbol erasures.

Steps:

1) Compute the $m \times (d-1)$ syndrome array
$$S = \left(\begin{array}{c|c|c|c} H_0 \Upsilon_0 & H_1 \Upsilon_1 & \ldots & H_{n-1} \Upsilon_{n-1} \end{array} \right) H_{\mathrm{GRS}}^{\mathsf{T}} .$$

2) Compute the modified syndrome array to be the unique $\varrho \times (d-1)$ matrix σ that satisfies the congruence
$$\sigma(y,x) \equiv S(y,x) \prod_{j \in \mathcal{K}} (1 - \alpha_j x) \pmod{\{x^{d-1}, y^\varrho\}} .$$

3) For every $\ell \in \langle \varrho \rangle$ do:

   a) Compute row $\varrho - 1$ in the unique $\varrho \times (d-1)$ matrix $\sigma^{(\ell)}$ that satisfies the congruence
   $$\sigma^{(\ell)}(y,x) \equiv \mathrm{B}^{(\ell)}(y)\, \sigma(y,x)\, (1 - \alpha_{j_\ell} x)$$
   $$(\mathrm{mod}\ \{x^{d-1}, y^\varrho\}) ,$$
   where $\mathrm{B}^{(\ell)}(y)$ is as in (30).

   b) Decode $e_{j_\ell}^{(\ell)}$ (i.e., entry j_ℓ in $\boldsymbol{e}^{(\ell)}$) by applying a decoder for $\mathcal{C}_{\mathrm{GRS}}$ using row $\varrho - 1$ in $\sigma^{(\ell)}$ as syndrome and assuming that columns indexed by $\mathcal{K} \cup \{j_\ell\}$ are erased. Compute $\varepsilon_{\kappa_\ell, j_\ell} = e_{j_\ell}^{(\ell)} \cdot \beta_{\kappa_\ell, j_\ell}^{1-\varrho}$.

   c) Update the received array Υ and the syndrome array S by
   $$\Upsilon(y,x) \leftarrow \Upsilon(y,x) - \varepsilon_{\kappa_\ell, j_\ell} \cdot x^{j_\ell} y^{\kappa_\ell}$$
   $$S(y,x) \leftarrow S(y,x) - \varepsilon_{\kappa_\ell, j_\ell} \cdot \mathrm{T}_{d-1}(x; \alpha_{j_\ell}) \cdot \mathrm{T}_m(y; \beta_{j_\ell}) .$$

4) For every $h \in \langle m \rangle$, apply a decoder for $\mathcal{C}_{\mathrm{GRS}}$ using row h of S as syndrome and assuming that columns indexed by $\mathcal{K}$ are erased. Let E be the $m \times n$ matrix whose rows are the decoded error vectors for all $h \in \langle m \rangle$.

5) Compute the error array
$$\mathcal{E} = \left(\begin{array}{c|c|c|c} H_0^{-1} E_0 & H_1^{-1} E_1 & \ldots & H_{n-1}^{-1} E_{n-1} \end{array} \right) .$$

Output:

- Decoded array $\Upsilon - \mathcal{E}$ of size $m \times n$.

Fig. 3. Decoding of errors and erasures of type (T1), (T2), (T4), but not of type (T3). (See Section IV-B.)

$j_0, j_1, \ldots, j_w$ are all distinct, while $j_w = j_{w+1} = \cdots = j_{\vartheta-1}$. Furthermore, ϑ and w should satisfy the inequalities
$$\vartheta \le \frac{m}{2} , \tag{33}$$
$$w + \tau + \rho \le d - 2 . \tag{34}$$

(While the inequality in (33) is already part of the requirements in Theorem 5, we need the second inequality so that (13) will hold. Specifically, the inequality in (34) says that the number of erroneous columns does not exceed $d - 1$. Observe that the inequality $2\tau + \rho \le d - 2$ and the inequality in (33), together imply the second inequality therein whenever $m \le d - \rho$.)

Without any loss of generality, we will also assume that $\varepsilon_{\kappa_\ell, j_\ell} \ne 0$ for every $\ell \in \langle \vartheta \rangle$. The set $\{j_\ell\}_{\ell \in \langle w+1 \rangle}$ will be denoted hereafter by $\mathcal{L}'$. When $\vartheta = 0$, we formally define w to be 0 and $\mathcal{L}'$ to be the empty set.

Let the modified syndrome σ be the unique $m \times (d-1)$

matrix that satisfies
$$\sigma(y,x) \equiv S(y,x) \cdot \prod_{j \in \mathcal{K}} (1 - \alpha_j x) \pmod{x^{d-1}} ,$$
and let $\tilde{S}$ be the $m \times (d-1-\rho)$ matrix formed by the columns of σ that are indexed by $\langle \rho, d-1 \rangle$. We recall from (13) that $\mu = \mathsf{rank}(\tilde{S}) = \mathsf{rank}((E)_{\mathcal{J} \cup \mathcal{L}'})$.

If $\mu \ge 2w + 2$, then we can regard the columns that are indexed by $\mathcal{L}'$ as full block errors (namely, errors of type (T1)), and the conditions of Lemma 3 will still be satisfied, namely, we will have
$$2(\tau + w + 1) + \rho \le d + \mu - 2 .$$

Therefore, we assume from now on that $\mu \le 2w + 1$.

By (13) we get that for every $j \in \mathcal{J} \cup \mathcal{L}'$, column E_j belongs to $\mathsf{colspan}(\tilde{S})$. In particular, this holds for $j \in \mathcal{L}' \setminus \{j_w\}$, in which case E_j (in polynomial notation) takes the form
$$E_j(y) = \varepsilon_{\kappa, j} \cdot \mathrm{T}_m(y; \beta_{\kappa, j}) .$$

Let the row vectors $\boldsymbol{a}_0, \boldsymbol{a}_1, \ldots, \boldsymbol{a}_{m-\mu-1}$ form a basis of the dual space of $\mathsf{colspan}(\tilde{S})$, and for every $i \in \langle m - \mu \rangle$, let $a_i(y)$ denote the polynomial of degree less than m with coefficient vector $\boldsymbol{a}_i$; we can further assume that this basis is in echelon form, i.e., $\deg a_0(y) < \deg a_1(y) < \ldots < \deg a_{m-\mu-1}(y) < m$. This, in turn, implies that the degree of $a(y) = \gcd(a_0(y), a_1(y), \ldots, a_{m-\mu-1}(y))$ satisfies
$$\deg a(y) \le \mu ,$$
namely, $a(y)$ has at most μ ($\le 2w+1$) distinct roots in F. Now, it is easy to see that for every $\xi \in F$, the column vector $(\xi^h)_{h \in \langle m \rangle}$ (also represented as $\mathrm{T}_m(y; \xi)$) belongs to $\mathsf{colspan}(\tilde{S})$ (and, hence, to $\mathsf{colspan}(E)_{\mathcal{J} \cup \mathcal{L}'}$), if and only if ξ is a root of $a(y)$. In particular, $\beta_{\kappa_\ell, j_\ell}$ is a root of $a(y)$ for every $\ell \in \langle w \rangle$. We denote by $\mathcal{R}$ the root subset
$$\mathcal{R} = \{(\kappa, j) : a(\beta_{\kappa, j}) = 0\} , \tag{35}$$
and define the polynomial $\mathrm{A}(y)$ by
$$\mathrm{A}(y) = \sum_{i=0}^{\eta} \mathrm{A}_i y^i = \prod_{(\kappa, j) \in \mathcal{R}} (1 - \beta_{\kappa, j} y) , \tag{36}$$
where $\eta = |\mathcal{R}|$.

Consider the $(m-\eta) \times n$ matrix $\hat{E} = (\hat{e}_{h,j})_{h \in \langle m-\eta \rangle, j \in \langle n \rangle}$ which is formed by the rows of $\mathrm{A}(y) E(y,x)$ that are indexed by $\langle \eta, m \rangle$. Specifically,
$$\hat{e}_{h,j} = \sum_{i=0}^{\eta} \mathrm{A}_i\, e_{h+\eta-i, j} , \quad h \in \langle m - \eta \rangle , \quad j \in \langle n \rangle$$
(compare with (32)). Respectively, let $\hat{S}$ be the $(m-\eta) \times (d-1-\rho)$ matrix formed by the rows of $\mathrm{A}(y) \tilde{S}(y,x)$ that are indexed by $\langle \eta, m \rangle$. It readily follows that $\hat{E}_{j_\ell}(y) = 0$ for $\ell \in \langle w \rangle$ and that
$$\hat{E}_{j_w}(y) = \sum_{\ell \in \langle w, \vartheta \rangle} \left(\varepsilon_{\kappa_\ell, j_w} \beta_{\kappa_\ell, j_w}^\eta \mathrm{A}(\beta_{\kappa_\ell, j_w}^{-1}) \right) \cdot \mathrm{T}_{m-\eta}(y; \beta_{\kappa_\ell, j_w}) .$$
$$\tag{37}$$

Observe that the number of summands on the right-hand side of (37) is $\vartheta - w$, and that number is bounded from above by $m - 2w - 1 \leq m - \mu \leq m - \eta$. This means that $\hat{E}_{j_w}(y) = 0$ if and only if $A(\beta_{\kappa_\ell, j_w}^{-1}) = 0$ for all $\ell \in \langle w, \vartheta \rangle$. We also recall that

$$\mathsf{rank}(\hat{S}) = \mathsf{rank}((\hat{E})_{\mathcal{J} \cup \{j_w\}}) = \mu - \eta . \tag{38}$$

Next, we distinguish between the following three cases.

Case 1: $\eta = \mu$. By (38) we must have $\hat{E}_{j_w}(y) = 0$, which is equivalent to having $A(\beta_{\kappa_\ell, j_\ell}^{-1}) = 0$ for all $\ell \in \langle \vartheta \rangle$. Thus, assuming this case, we have $\mathcal{L} \subseteq \mathcal{R}$, and the decoding problem then reduces to the one discussed in Section IV-B.

Case 2: $\eta = \mu - 1$. If $\hat{E}_{j_w}(y) = 0$ then $\mathcal{L} \subseteq \mathcal{R}$. Otherwise, it follows from (38) that each column in $\hat{S}$ must be a scalar multiple of $\hat{E}_{j_w}$. The entries of $\hat{E}_{j_w}$, in turn, form a sequence that satisfies the (shortest) linear recurrence

$$B(y) = \sum_{i=0}^{|\mathcal{R}'|} B_i y^i = \prod_{(\kappa, j) \in \mathcal{R}'} (1 - \beta_{\kappa, j} y) ,$$

where

$$\mathcal{R}' = \left\{ (\kappa_\ell, j_w) : \ell \in \langle w, \vartheta \rangle \text{ and } A(\beta_{\kappa_\ell, j_w}^{-1}) \neq 0 \right\} .$$

Indeed, this recurrence is uniquely determined, since the number of entries in $\hat{E}_{j_w}$, which is $m - \eta = m - \mu + 1 \geq m - 2w$, is at least twice the degree $|\mathcal{R}'|$ ($\leq \vartheta - w$) of $B(y)$. The recurrence can be computed efficiently from any nonzero column of $\tilde{S}$ using Massey's algorithm [21]. We now have $\mathcal{L} \subseteq \mathcal{R} \cup \mathcal{R}'$, where

$$\begin{aligned}
|\mathcal{R} \cup \mathcal{R}'| &= |\mathcal{R}| + |\mathcal{R}'| \\
&\leq \eta + \vartheta - w \\
&\leq 2w + \vartheta - w \\
&\leq \vartheta + w \\
&< m .
\end{aligned}$$

So the decoding problem again reduces to that in Section IV-B.

Case 3: $\eta \leq \mu - 2$. If $\hat{E}_{j_w}(y) = 0$ then (again) $\mathcal{L} \subseteq \mathcal{R}$. Otherwise, the conditions of Lemma 3 hold with respect to $\hat{E}$ and to the matrix $\hat{Z}$ formed by the rows of $A(y)Z(y, x)$ indexed by $\langle \eta, m \rangle$ (each such row is a codeword of $\mathcal{C}_{\mathrm{GRS}}$). Hence, we can decode $\hat{E}$. Next, we observe from (37) that for $j = j_w$, the vector $\hat{E}_j(y)$ can be seen as a syndrome of the column vector

$$\mathcal{E}_j^*(y) = \sum_{\kappa \in \langle m \rangle : A(\beta_{\kappa, j}^{-1}) \neq 0} \varepsilon_{\kappa, j} y^\kappa$$

with respect to the following $(m - \eta) \times m$ parity-check matrix of a GRS code:

$$H_{\mathrm{GRS}}^{(j)} = \left(v_{\kappa, j} \beta_{\kappa, j}^h \right)_{h \in \langle m - \eta \rangle, \kappa \in \langle m \rangle} , \tag{39}$$

where

$$v_{\kappa, j} = \begin{cases} \beta_{\kappa, j}^\eta A(\beta_{\kappa, j}^{-1}) & \text{if } A(\beta_{\kappa, j}^{-1}) \neq 0 \\ 1 & \text{otherwise} \end{cases} . \tag{40}$$

And since the Hamming weight of $\mathcal{E}_j^*$ is at most $\vartheta - w < (m - \eta)/2$, we can decode $\mathcal{E}_j^*$ uniquely from $\hat{E}_j$ (again, under

the running assumption that $j = j_w$). Thus, for every κ such that $A(\beta_{\kappa, j}^{-1}) \neq 0$, we can recover the error value $\varepsilon_{\kappa, j}$ and subtract it from the respective entry of Υ, thereby making $\mathcal{R}$ a superset of the remaining symbol errors. The problem though is that we do not know the index j_w. Therefore, we apply the above process to *every* nonzero column in $\hat{E}$ with index $j \notin \mathcal{K}$. A decoding failure means that j is certainly not j_w, and a decoding success for $j \neq j_w$ will just cause us to incorrectly change already corrupted-columns in Υ, without introducing new erroneous columns. We can then proceed with the decoding of Υ as in Section IV-B.

Figure 4 presents the implied decoding algorithm of a combination of errors of type (T1), (T2), and (T3), provided that the type-(T3) errors satisfy the assumptions laid out at the beginning of this section; as said earlier, these assumptions hold when $m \leq d - \rho$ and the number of type-(T3) errors is at most 3. Steps 1–3, 6a, and 7–8 in Figure 4 are essentially applications of steps in Figures 2 and 3. Next, we analyze the complexity of the remaining steps in Figure 4, staring with Step 4a. A basis in echelon form of the left kernel of $\tilde{S}$ can be found using $O(d^2 m)$ operations in F, and from this basis we can compute $a(y)$ using m applications of Euclid's algorithm, amounting to $O(dm^2)$ operations. The set $\mathcal{R}$ can then be found in Step 4b via Chien search, requiring $O(mn \cdot \min(d, m))$ operations, followed by $O(d^2 m)$ operations to compute the matrix $\hat{S}$ in Step 4c. The complexity of Step 5 is dictated by Step 5b therein which, with Chien search, can be implemented using $O(m^2 n)$ operations. Finally, Step 6b requires $O(d(d + m)m)$ operations in F. In summary, the decoding complexity of the algorithm in Figure 4 amounts to $O((d + m)mn)$ operations for syndrome computation and the Chien search, and $O(d(d + m)m)$ operations for the other steps.

APPENDIX A
ANALYSIS FOR EXAMPLE 14

We show that (18) implies the inequality (17). Specifically, we assume the following value of $\mathsf{red}_q(N, D)$

$$\mathsf{red}_q(N, D) = \lceil \log_q(V_q(N - 1, D - 2) + 1) \rceil ,$$

where

$$V_q(N - 1, D - 2) = \sum_{i=0}^{D-2} \binom{N-1}{i} (q - 1)^i ,$$

and show that, for $1 < D_1 < D_2 \leq \frac{q-1}{2q-1} N + 2$,

$$\mathsf{red}_q(N, D_2) \geq \mathsf{red}_q(N, D_1) + (D_2 - D_1) .$$

To this end, it suffices to show that for $1 < D \leq \frac{q-1}{2q-1} N + 1$,

$$\mathsf{red}_q(N, D + 1) \geq \mathsf{red}_q(N, D) + 1 . \tag{41}$$

For the above range of D we have

$$\frac{N - i}{i} \geq \frac{N - D + 1}{D - 1} \geq \frac{q}{q - 1} , \quad 1 \leq i \leq D - 1 ,$$

Input:

- Array Υ of size $m \times n$ over F.
- Set $\mathcal{K}$ of indexes of column erasures.

Steps:

1) Compute the $m \times (d-1)$ syndrome array

$$S = \left(\ H_0\Upsilon_0 \mid H_1\Upsilon_1 \mid \ldots \mid H_{n-1}\Upsilon_{n-1}\ \right) H_{\mathrm{GRS}}^{\mathsf{T}}\ .$$

2) Compute the $m \times (d-1-\rho)$ matrix $\tilde{S}$ formed by the columns of $S(y,x) \prod_{j \in \mathcal{K}} (1-\alpha_j x)$ that are indexed by $\langle \rho, d-1 \rangle$. Let $\mu = \mathsf{rank}(\tilde{S})$.

3) (Attempt to correct assuming $|\mathcal{L}'| \leq \mu/2$.) Apply Steps 3–4 in Figure 2 (with $K = \mathcal{K}$) to the modified syndrome array $\sigma(y,x)$, to produce an error array E. If decoding is successful, go to **Step 8**.

4) a) Compute the greatest common divisor $a(y)$ of a basis of the left kernel of $\tilde{S}$.

   b) Compute the set $\mathcal{R}$ and the polynomial $\mathrm{A}(y)$ as in (35)–(36). Let $\eta = |\mathcal{R}|$.

   c) Compute the $(m-\eta) \times (d-1-\rho)$ matrix $\hat{S}$ formed by the rows of $\mathrm{A}(y)\tilde{S}(y,x)$ that are indexed by $\langle \eta, m \rangle$.

5) If $\eta = \mu - 1$ then do:

   a) Compute the shortest linear recurrence $\mathrm{B}(y)$ of any nonzero column in $\hat{S}$.

   b) Compute the set

   $$\mathcal{R}' = \left\{ (\kappa, j)\ :\ \mathrm{A}(\beta_{\kappa,j}^{-1}) \neq 0 \text{ and } \mathrm{B}(\beta_{\kappa,j}^{-1}) = 0 \right\}.$$

   c) If $|\mathcal{R}'| = \deg \mathrm{B}(y)$ and $|\mathcal{R}'| \leq m - \eta$ then update $\mathcal{R} \leftarrow \mathcal{R} \cup \mathcal{R}'$.

6) Else if $\eta \leq \mu - 2$ then do:

   a) Apply Steps 2–4 in Figure 2 (with $K = \mathcal{K}$) to the syndrome array $\hat{S}$, to produce an error array $\hat{E}$.

   b) For every index $j \notin \mathcal{K}$ of a nonzero column of $\hat{E}$ do:

   i) Apply a decoder for the GRS code with the parity-check matrix $H_{\mathrm{GRS}}^{(j)}$ as in (39)–(40), with $\hat{E}_j$ as syndrome, to produce an error vector $\mathcal{E}_j^*$.

   ii) If decoding in Step 6(b)i is successful then let $E_j^* = H_j\mathcal{E}_j^*$ and update $\Upsilon_j \leftarrow \Upsilon_j - \mathcal{E}_j^*$ and $S(y,x) \leftarrow S(y,x) - E_j^*(y) \cdot \mathrm{T}_{d-1}(x; \alpha_{j_\ell})$.

7) Apply Steps 2–4 in Figure 3 to S, $\mathcal{K}$, and $\mathcal{R}$, to produce an error array E.

8) Compute the error array

$$\mathcal{E} = \left(\ H_0^{-1}E_0 \mid H_1^{-1}E_1 \mid \ldots \mid H_{n-1}^{-1}E_{n-1}\ \right)\ .$$

Output:

- Decoded array $\Upsilon - \mathcal{E}$ of size $m \times n$.

Fig. 4. Decoding of errors and erasures of type (T1), (T2), (T4), and with restrictions on errors of type (T3). (See Section IV-C.) For the sake of simplicity, we assume that there are no erasures of type (T4).

which readily implies that

$$\binom{N-1}{i} \geq \frac{q}{q-1} \cdot \binom{N-1}{i-1}, \quad 1 \leq i \leq D-1\ ,$$

and a simple induction on i thus yields

$$\binom{N-1}{D-1} \geq \left(\frac{q}{q-1}\right)^i \cdot \binom{N-1}{D-1-i}, \quad 0 \leq i \leq D-1\ . \quad (42)$$

It follows that (for the above range of D),

$$\mathsf{V}_q(N-1, D-2) = (q-1)^{D-1} \sum_{i=1}^{D-1} \binom{N-1}{D-1-i}(q-1)^{-i}$$

$$\overset{(42)}{\leq} \binom{N-1}{D-1}(q-1)^{D-1} \sum_{i=1}^{D-1} q^{-i}$$

$$< \binom{N-1}{D-1}(q-1)^{D-2}\ .$$

Namely,

$$\mathsf{V}_q(N-1, D-2) + 1 \leq \binom{N-1}{D-1}(q-1)^{D-2}\ .$$

Multiplying both sides by $q-1$ yields

$$(q-1) \cdot (\mathsf{V}_q(N-1, D-2) + 1) \leq \binom{N-1}{D-1}(q-1)^{D-1}$$

and, so,

$$q \cdot (\mathsf{V}_q(N-1, D-2) + 1)$$
$$\leq \mathsf{V}_q(N-1, D-2) + 1 + \binom{N-1}{D-1}(q-1)^{D-1}$$
$$= \mathsf{V}_q(N-1, D-1) + 1\ .$$

We conclude that

$$1 + \log_q(\mathsf{V}_q(N-1, D-2)+1) \leq \log_q(\mathsf{V}_q(N-1, D-1)+1)\ ,$$

and (41) finally follows by taking ceilings on both sides.

APPENDIX B
ANALYSIS FOR EXAMPLE 16

We derive the lower bound (25) on the expression (24):

$$\sum_{i=1}^{v}(\mathsf{r}_i - \mathsf{r}_{i-1})(\mathsf{D}_{i-1} - 1 - 2\tau)$$

$$\overset{(1)}{\geq} \sum_{i=1}^{v}(\mathsf{r}_i - \mathsf{r}_{i-1})\left(\left\lceil \frac{2\vartheta + 1}{\mathsf{d}_{i-1}} \right\rceil - 1\right)$$

$$\geq \sum_{i=1}^{v}(\mathsf{r}_i - \mathsf{r}_{i-1})\left(\left\lceil \frac{2\vartheta + 1}{\mathsf{r}_{i-1} + 1} \right\rceil - 1\right)$$

$$= \sum_{i=1}^{v} \sum_{j=\mathsf{r}_{i-1}+1}^{\mathsf{r}_i}\left(\left\lceil \frac{2\vartheta + 1}{\mathsf{r}_{i-1} + 1} \right\rceil - 1\right)$$

$$\geq \sum_{j=1}^{\mathsf{r}_v}\left(\left\lceil \frac{2\vartheta + 1}{j} \right\rceil - 1\right)$$

$$\geq \sum_{j=1}^{2\vartheta}\left(\left\lceil \frac{2\vartheta + 1}{j} \right\rceil - 1\right)$$

$$= \sum_{j=1}^{\vartheta}\left\lceil \frac{2\vartheta + 1}{j} \right\rceil\ ,$$

where the penultimate step follows from $\mathsf{r}_v \geq 2\vartheta$. The lower bound (25) immediately follows by the known expression for harmonic sums [10, p. 264].

REFERENCES

[1] K.A.S. ABDEL-GHAFFAR AND M. HASSNER, *Multilevel error-control codes for data storage channels, IEEE Trans. Inf. Theory,* 37 (1991) 735–741.

[2] M. BLAUM, J.L. HAFNER, AND S. HETZLER, *Partial-MDS codes and their application to RAID type of architectures,* IBM Res. Rep. No. RJ10498 (February 2012).

[3] D. BLEICHENBACHER, A. KIAYAS, AND M. YUNG, *Decoding interleaved Reed–Solomon codes over noisy channels, Theor. Comput. Sci.,* 379 (2007), 348–360.

[4] E.L. BLOKH AND V.V. ZYABLOV, *Coding of generalized concatenated codes, Probl. Inf. Transm.,* 10 (1974), 218–222.

[5] A. BROWN, L. MINDER, AND A. SHOKROLLAHI, *Probabilistic decoding of interleaved RS-codes of the Q-ary symmetric channel, Proc. IEEE Int. Symp. Inf. Theory,* Chicago, IL, 2004, pp. 326.

[6] D. COPPERSMITH AND M. SUDAN, *Reconstructing curves in three (and higher) dimensional space from noisy data, Proc. 35th Annual ACM Symp. Theory of Computing,* San Diego, CA, 2003, pp. 136-142.

[7] I. DUMER, *Concatenated codes and their multilevel generalizations,* in *Handbook of Coding Theory, Volume II,* V.S. Pless, W.C. Huffman (Editors), North-Holland, Amsterdam, 1998, pp. 1911–1988.

[8] G.D. FORNEY, JR., *On decoding BCH codes, IEEE Trans. Inf. Theory,* 11 (1965) 549–557.

[9] G.-L. FENG AND K.K. TZENG, *A generalization of the Berlekamp–Massey algorithm for multisequence shift-register synthesis with applications to decoding cyclic codes, IEEE Trans. Inf. Theory,* 37 (1991), 1274–1287.

[10] R.L. GRAHAM, D.E. KNUTH, AND O. PATASHNIK, *Concrete Mathematics,* Addison-Wesley, Reading, MA, 1989.

[11] V. GURUSWAMI AND M. SUDAN, *Improved decoding of Reed–Solomon and algebraic–geometry codes, IEEE Trans. Inf. Theory,* 45 (1999), 1757–1767.

[12] C. HASLACH AND A.J.H. VINCK, *A decoding algorithm with restrictions for array codes, IEEE Trans. Inf. Theory,* 45 (1999), 2339–2344 (and correction in the same publication, 47 (2001), 479).

[13] C. HASLACH AND A.J.H. VINCK, *Efficient decoding of interleaved linear block codes, Proc. IEEE Int. Symp. Inf. Theory,* Sorrento, Italy, 2000, p. 149.

[14] J. JUSTESEN, C. THOMMESEN, AND T. HØHOLDT, *Decoding of concatenated codes with interleaved outer codes, Proc. IEEE Int. Symp. Inform. Theory,* Chicago, IL, 2004, p. 328.

[15] M. KASAHARA, S. HIRASAWA, Y. SUGIYAMA, AND T. NAMEKAWA, *New classes of binary codes constructed on the basis of concatenated codes and product codes, IEEE Trans. Inf. Theory,* 22 (1976), 462–467.

[16] R. KOETTER AND A. VARDY, *The re-encoding transformation in algebraic list-decoding of Reed–Solomon codes, IEEE Trans. Inf. Theory,* 57 (2011), 633–647.

[17] V.YU. KRACHKOVSKY, AND Y.X. LEE, *Decoding of parallel Reed–Solomon codes with applications to product and concatenated codes, Proc. IEEE Int. Symp. Inf. Theory,* Cambridge, MA, 1998, p. 55.

[18] H. KURZWEIL, M. SEIDL, AND J.B. HUBER, *Reduced-complexity collaborative decoding of interleaved Reed–Solomon and Gabidulin codes, Proc. IEEE Int. Symp. Inf. Theory,* St. Petersburg, Russia, 2011, pp. 2557–2561.

[19] S. LIN AND D.J. COSTELLO, JR., *Error Control Coding: Fundamentals and Applications,* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1983.

[20] F.J. MACWILLIAMS AND N.J.A. SLOANE, *The Theory of Error-Correcting Codes,* North-Holland, Amsterdam, 1977.

[21] J.L. MASSEY, *Shift-register synthesis and BCH decoding, IEEE Trans. Inf. Theory,* 15 (1969), 122–127.

[22] J.J. METZNER AND E.J. KAPTUROWSKI, *A general decoding technique applicable to replicated file disagreement location and concatenated code decoding, IEEE Trans. Inf. Theory,* 36 (1990), 1274–1287.

[23] W.W. PETERSON AND E.J. WELDON, JR., *Error-Correcting Codes,* MIT Press, Cambridge, MA, 1972.

[24] R.M. ROTH, *Introduction to Coding Theory,* Cambridge Univ. Press, Cambridge, UK, 2006.

[25] R.M. ROTH AND G. SEROUSSI, *Reduced-redundancy product codes for burst error correction, IEEE Trans. Inf. Theory,* 44 (1998), 1395–1406.

[26] S. SAKATA, *Decoding binary 2-D Cyclic codes by the 2-D Berlekamp–Massey algorithm, IEEE Trans. Inf. Theory,* 37 (1991), 1200–1203.

[27] G. SCHMIDT, V.R. SIDORENKO, AND M. BOSSERT, *Collaborative decoding of interleaved Reed–Solomon codes and concatenated code designs, IEEE Trans. Inf. Theory,* 55 (2009), 2991–3011.

[28] L. STORME AND J.A. THAS, *M.D.S. codes and arcs in $PG(n, q)$ with q even: an improvement of the bounds of Bruen, Thas, and Blokhuis, J. Comb. Theory A,* 62 (1993), 139–154.

[29] A. WACHTER–ZEH, A. ZEH, AND M. BOSSERT, *Decoding interleaved Reed–Solomon codes beyond their joint error-correcting capability, Des. Codes Cryptogr.,* to appear.

[30] Y. WU, *New list decoding algorithms for Reed–Solomon and BCH codes, IEEE Trans. Inf. Theory,* 54 (2008), 3611–3630.

[31] J. WU AND D.J. COSTELLO, JR., *New multilevel codes over $\mathrm{GF}(q)$, IEEE Trans. Inf. Theory,* 38 (1992), 933–939.

[32] V.A. ZINOV'EV, *Generalized concatenated codes for channels with error bursts and independent errors, Probl. Inf. Transm.,* 17 (1981), 254–260.

[33] V.A. ZINOV'EV AND V. ZYABLOV, *Correction of error bursts and independent errors using generalized cascade codes, Probl. Inf. Transm.,* 15 (1979), 125–134.